Doctoral Thesis

Deep Learning for Laser Materials Processing and Self-Piercing Riveting

Sehyeok Oh

Department of Mechanical Engineering

Graduate School of UNIST

2020

Deep Learning for Laser Materials Processing and Self-Piercing Riveting

Sehyeok Oh

Department of Mechanical Engineering

Graduate School of UNIST

Deep Learning for Laser Materials Processing and Self-Piercing Riveting

A dissertation
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Sehyeok Oh

June/19/2020
Approved by

_____

Advisor
Hyungson Ki

Deep Learning for Laser Materials Processing and Self-Piercing Riveting

Sehyeok Oh

This certifies that the dissertation of Sehyeok Oh is approved.

June/19/2020

signature

_____

Advisor: Hyungson Ki

signature

_____

Jaeseon Lee: Committee Member #1

signature

_____

Jaesung Jang: Committee Member #2

signature

_____

Jun Choi: Committee Member #3

signature

_____

Sangwoo So: Committee Member #4

Abstract

Along with a worldwide increasing popularity of deep learning in computer science that began in the 2010s, it has been actively applied in the field of mechanical engineering as well, such as in computational fluid dynamics (CFD) simulation, topological design, and materials processing. Unlike the conventional numerical method of solving governing differential equations (physics-based), deep learning has presented a completely new perspective of analyzing modern technologies. Trained from a given dataset (experimental or simulation), the deep learning model can predict the future with very good accuracy, by spontaneously discovering intrinsic patterns contained in the data (thus, data-driven). In this dissertation, we present novel frameworks to make accurate predictions in three modern materials processing technologies (*i.e.*, laser heat treatment, laser keyhole welding, and self-piercing riveting), by applying a state-of-the-art deep learning architecture. We anticipate that the proposed deep learning frameworks will be an important milestone for the future advanced manufacturing applications using an artificial intelligence (AI).

In chapter 1, we introduced backgrounds of the three aforesaid materials processing technologies and deep learning, respectively. In the deep learning section, the focus was primarily placed on the algorithms of actively employed in computer vision, that is a convolutional neural network (CNN) for image recognition and a generative adversarial network (GAN) for image generation, which were the two main source frameworks adopted in this study.

In chapter 2, we proposed a deep learning-based hardness predictive model in laser surface hardening (heat treatment) of AISI H13 tool steel, from an input of cross-sectional temperature distribution (**the first deep learning model in laser hardening**). The objective of laser hardening is to improve the metal surface by locally enhancing the surface hardness, and the employed deep learning model succeeded in accurately predicting the amount of hardening on entire cross-section. For the model input, finite element method (FEM)-simulated cross-sectional temperature profile was used when the surface temperature reaches the maximum, and the model was based on a conditional generative adversarial network (cGAN) with the CNN encoder-decoder, which is a specialized structure in image-to-image translation (temperature-to-hardness translation in our model). The presented deep learning architecture is expected to be useful in a development of highly accurate process predicting systems in laser heat treatment.

In chapter 3, we studied a cross-section weld bead image prediction in laser keyhole welding of AISI 1020 steel, using state-of-the-art deep learning algorithms (**the first deep learning model in laser weld bead image prediction**). Predicting the bead shape has always been a challenging issue in laser keyhole welding, as the complex multi-physics phenomena come into play with high interfacial forces such as capillary and thermocapillary forces and recoil pressure. With our deep learning model, not only

the geometrical bead shape, but also a high-resolution optical microscopic (OM) weld bead image can be produced including keyhole, heat affected zone, substrate, porosity, and microstructures, from the two input parameters of laser intensity and beam scanning speed. The proposed deep learning model consisted of two successive generators which both exhibit an encoder-decoder structure based on the CNN. Additionally, in the second generator, multi-scale cGAN architecture was employed with deep residual connections, considering size of the OM image (high-resolution). We expect the presented deep learning framework to play a leading role in the future advanced modeling of laser keyhole welding.

In chapter 4, we presented a deep learning framework for predicting cross-sectional shape in self-piercing riveting (SPR) joining process (**the first deep learning model in SPR**). SPR process is getting popular in the automotive industry, as it can easily combine two or more sheets in a single step regardless of the material types (even can be applied to the dissimilar sheets such as steel–nonferrous metal and composite–metal). The quality of the SPR joint is determined by the cross-sectional shape, so its prediction is essential, which was conventionally carried out by the FEM simulation. Using our predictive model, without any concerns about the mesh and time step, highly accurate cross-sectional shape can be generated from a scalar input of punch force, within a few seconds. The proposed predictive model was a novel CNN-based deep residual generator in the cGAN architecture. The model presented in this dissertation opens up the possibilities of deep learning applications to the SPR process for the first time, and we anticipate that our model will play a central role in a development of future sophisticated AI models.

This Page Intentionally Left Blank

Contents

List of Figures

Chapter 1

Chapter 2

Chapter 3

Figure 3.18 Feature map comparison between the training set (1800 W and 5.313 mm/s, left figure) and test set (1273 W and 15.03 mm/s, right figure). The feature maps were extracted from the last second layer in the local enhancer network.

Chapter 4

List of Tables

Chapter 1. Introduction

1.1 Introduction of Modern Materials Processing

1.1.1 General Introduction of Modern Materials Processing

The modern materials processing technology includes high precision processing, next-generation materials processing, dissimilar materials processing, high-performance smart computing for the process modeling, 3D printing, multi-scale manufacturing, etc. Specifically, in this dissertation, we introduce three representative modern materials processing techniques for a deep learning application, which are laser heat treatment, laser keyhole welding, and self-piercing riveting (SPR) processes (**Figure 1.1**).

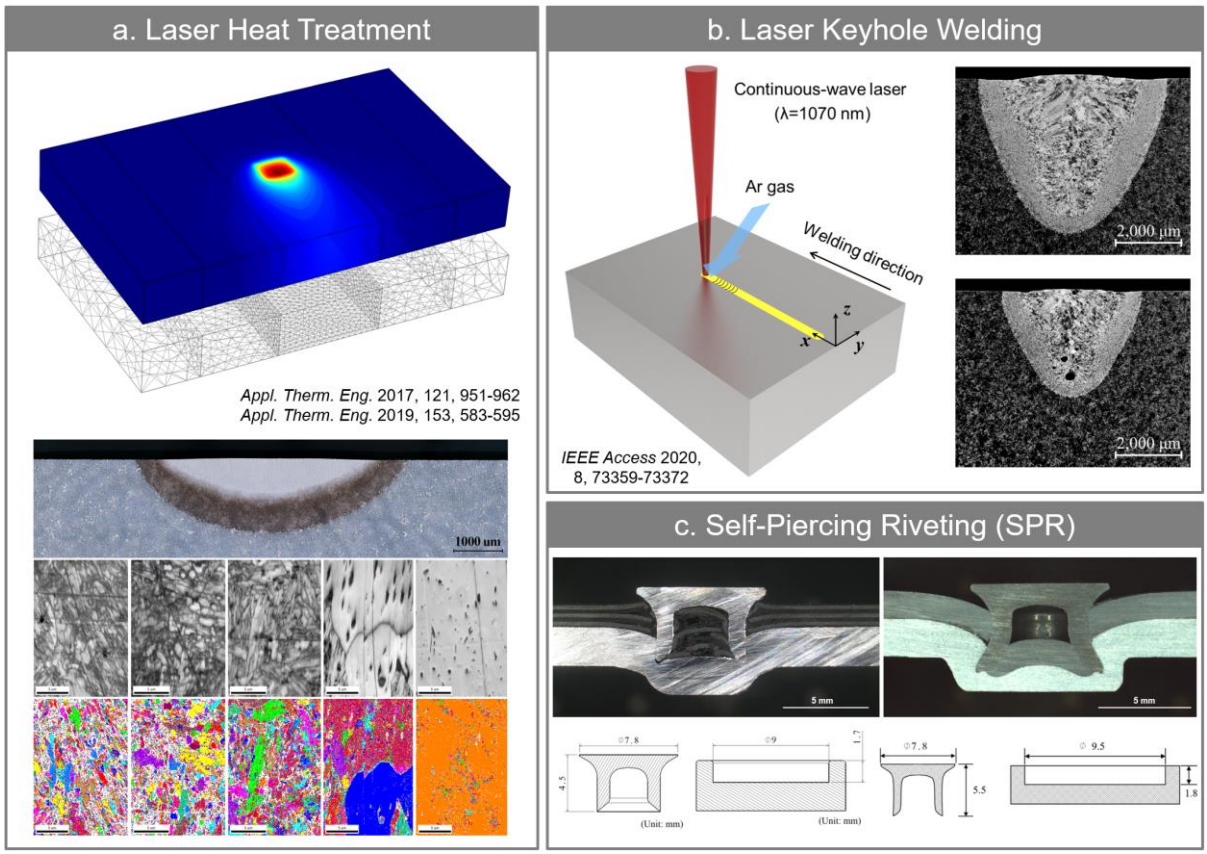

Figure 1.1. Introduction of three materials processing techniques conducted in this dissertation: (a) laser heat treatment; (b) laser keyhole welding; (c) self-piercing riveting. Reprinted with permission from [1-3] respectively. Copyright © 2017 Elsevier, Copyright © 2019 Elsevier, and Copyright © 2020 IEEE, respectively.

1.1.2 Laser Materials Processing and Laser Heat Treatment

In laser materials processing, two types of process regime exist in terms of time scale – thermal regime and nonthermal regime [4]. When laser beam is irradiated on material, surface electrons oscillate by external electric field from the laser, and generated thermal energy is transferred to near nuclei. Then thermal equilibrium is reached between the electron and nucleus, within 1~10 ps (**Figure 1.2**).

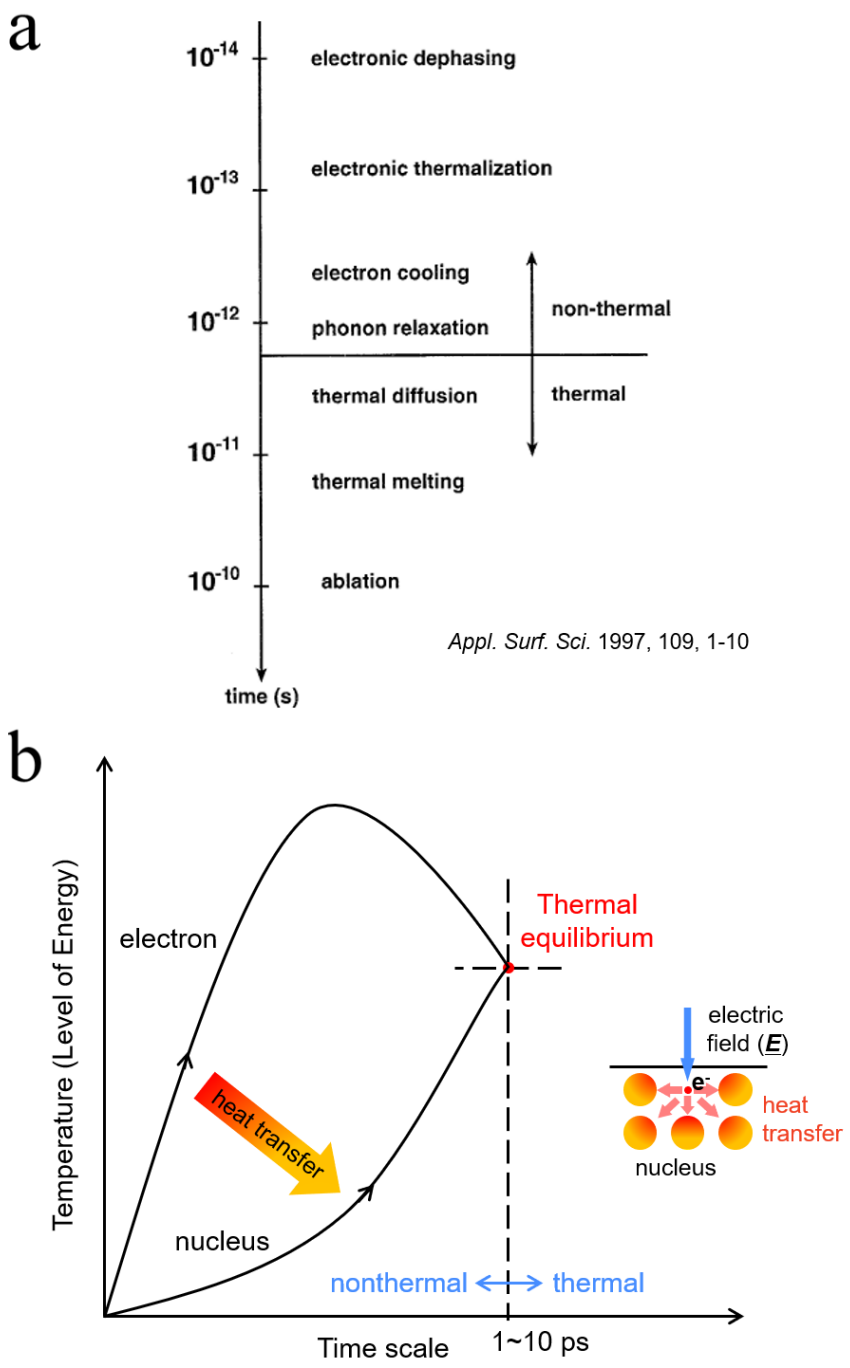

Figure 1.2. (a) Time scale for diverse secondary processes. Reprinted with permission from [4]. Copyright © 1997 Elsevier. (b) Schematic for thermal equilibrium between the electron and nucleus.

In the case of nonthermal regime (time scale smaller than 1~10 ps), the material is processed (*e.g.* bonding break) before thermal equilibrium, thus ultra-precision processing without thermal defects is possible [5]. On the other hand, in thermal regime (time scale greater than ~10 ps), conventional macroscopic processing is conducted, using laser beam as an external heat source. The various laser processing techniques in the thermal regime are presented in **Figure 1.3** [6].

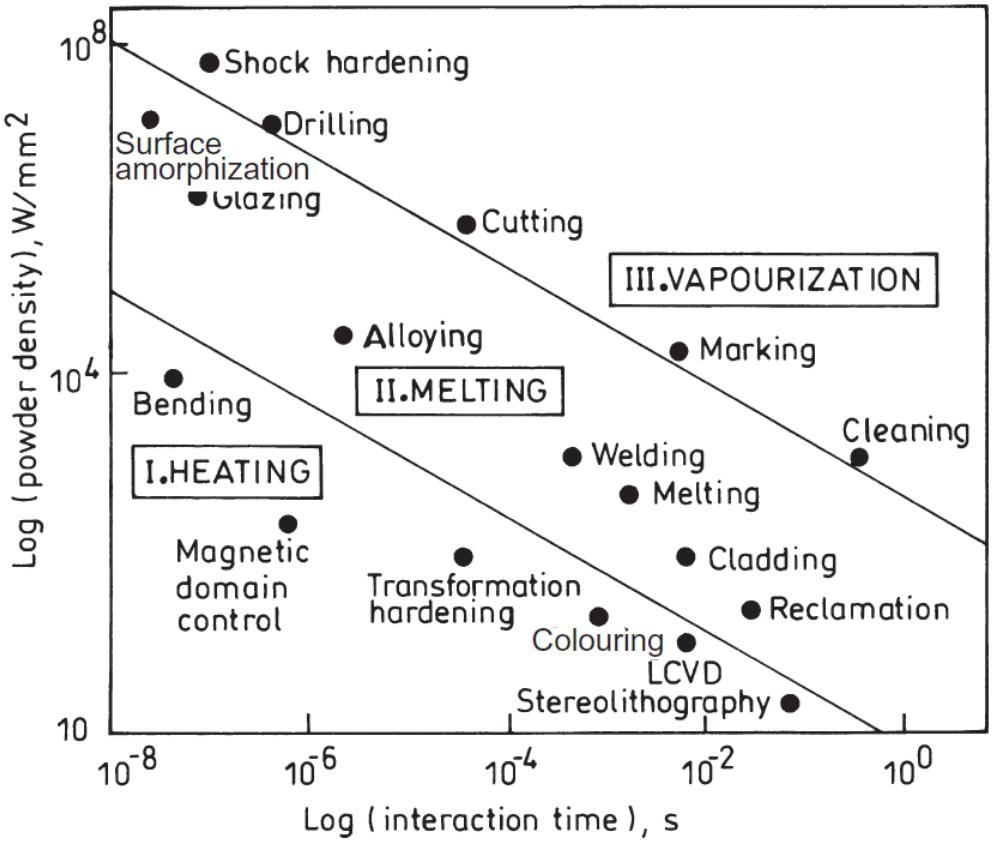

Figure 1.3. Schematic laser process map in thermal regime on a laser intensity–interaction time plane. Reprinted with permission from [6]. Copyright © 2003 Springer Nature.

Owing to its superb nature (*i.e.* super-precision; extremely high and uniform energy density; easy control of energy input by changing laser intensity and interaction time), laser beam has been widely employed as a heat source in materials processing, as presented in **Figure 1.3**. Especially, laser heat treatment (also known as laser transformation hardening) has attracted a large attention as a technique to improve strength, hardness, and wear resistance of steel parts and dies [1]. The quality of hardening is excellent compared with other hardening techniques thanks to the fast cooling rate (easy formation of martensite) [7]. Using the technique, an accurate prediction model for the process is necessary from the inputs of laser processing parameters (laser intensity and beam interaction time), to achieve a desired hardening extent.

However, due to the complexity of the process which multi-physics phenomena come into play (*i.e.* processes of thermal (energy transfer), metallurgical (solid-state phase transformation), and mechanical (stress development) are strongly coupled, with possible phase changes), development of a pure physics-based predictive model has undergone difficulties (it will also require high computational cost). In this reason, many researchers have tried to develop a simplified prediction model based on thermal characteristics appeared in laser hardening. Hardening is caused by brittle and hard microstructure of martensite, which is formed by rapid cooling of austenite such that carbon atoms do not have time to diffuse out of the crystal. For a decent hardening, following two conditions are required [8, 9]: (1) time exposure above the A3 temperature must be sufficient for carbon to fully diffuse in austenite, and (2) fast cooling must be guaranteed for as many as martensite formation. Quantifying both conditions, recently, Ki et al. [1, 9-12] reported effective carbon diffusion time (ECDT) and effective cooling time (ECT) based approaches in laser hardening (**Figure 1.4**).

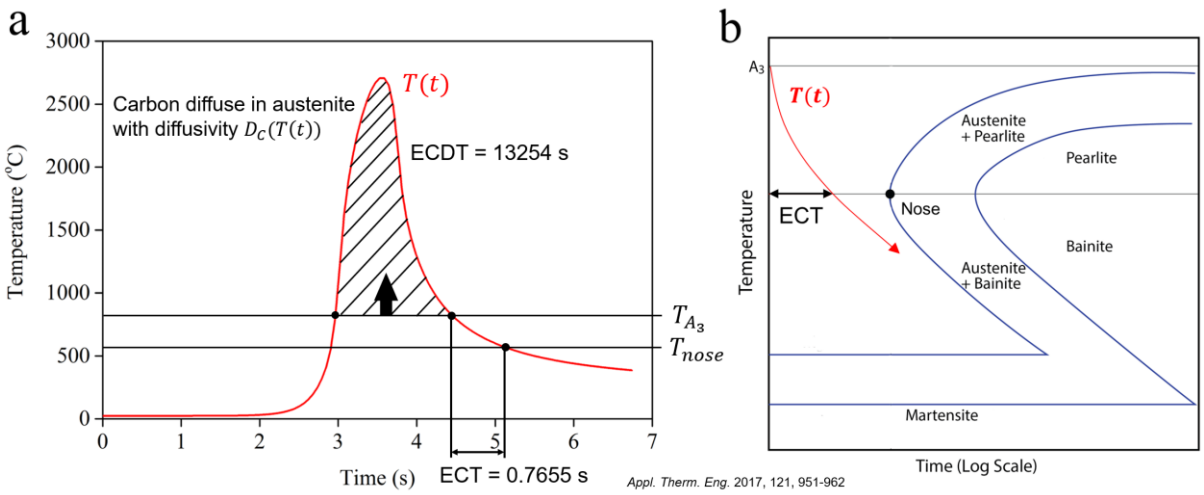

Figure 1.4. (a) Temperature-time curve in laser heat treatment with ECDT and ECT configurations (laser intensity and interaction time were 5884.5 W/cm^2 and 0.143 s, respectively). Reprinted with permission from [1]. Copyright © 2017 Elsevier. (b) Time-Temperature-Transformation (TTT) diagram for carbon steels. Reprinted with permission from [1]. Copyright © 2017 Elsevier.

The carbon diffusivity in austenite is a function of temperature, thus the actual temperature history must be considered to calculate more meaningful carbon diffusion time. In this context, the ECDT was calculated by integrating carbon diffusivity normalized by the diffusivity at A3 (when $T > T_{A3}$) [9]:

$$\mathrm{ECDT} = \frac{1}{D_c\left(T_{A_3}\right)} \int_{T>T_{A_3}} D_c(T(t))dt\,, \tag{1.1}$$

where D_c is the temperature-dependent carbon diffusivity, which is given by [13]

$$D_c(T) = 4.53\times10^{-7}\left(1+y_c(1-y_c)\frac{8339.9}{T}\right)\exp\left[-\left(\frac{1}{T}-2.221\times10^{-4}\right)(17767-26436y_c)\right]. \quad (1.2)$$

y_c was defined as $x_c/(1–x_c)$, where x_c is the mole fraction of carbon. ECT was defined as the time taken from A3 temperature to the nose temperature:

$$\mathrm{ECT} = t(T_{nose}) - t(T_{A_3}). \quad (1.3)$$

Note that the A1, A3, nose, and melting temperatures of typical carbon steels are 727, 793, 549, and 1470 °C, respectively [14, 15]. Based on the 3-D thermal simulation, Oh and Ki recently proposed predictive models in laser heat treatment of AISI H13 tool steel, by correlating ECDT–hardness and ECT–deflection angle [1], and by employing deep learning architecture [2] that will be discussed in **chapter 2** in detail (**Figure 1.5**).

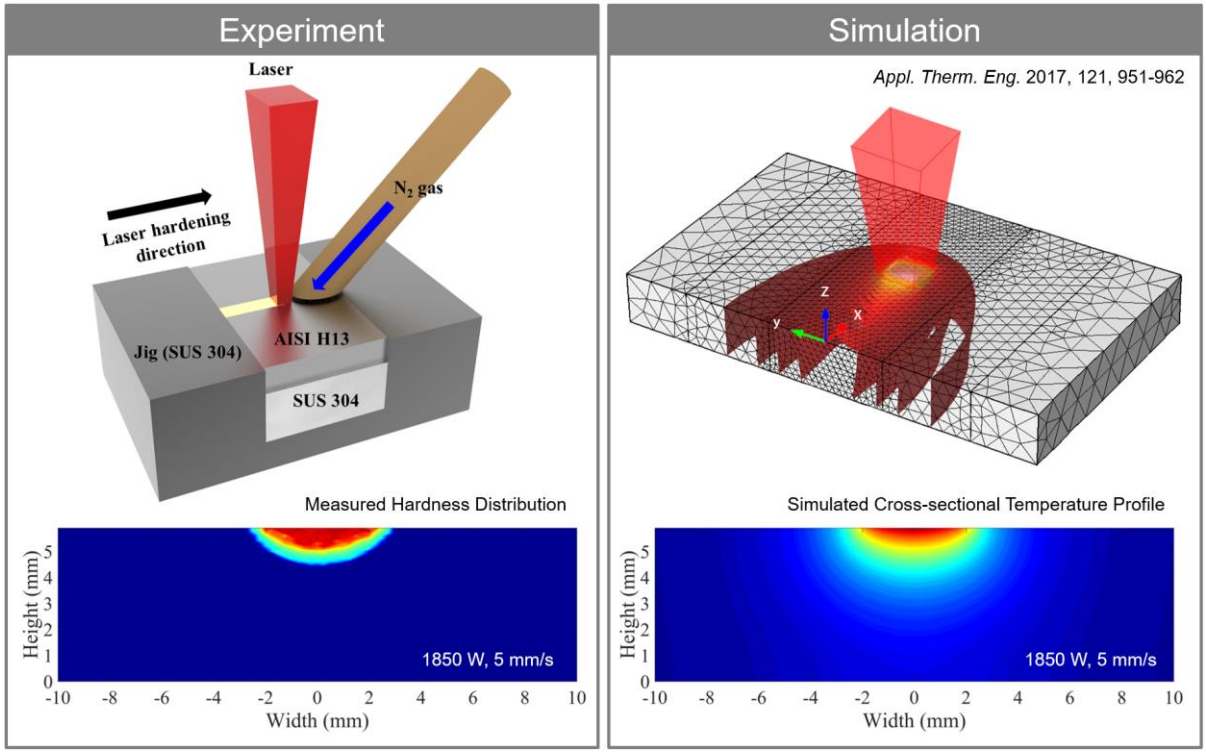

Figure 1.5. Schematics for the experiment (left) and thermal simulation (right) of the laser heat treatment. Measured cross-sectional hardness distribution and simulated temperature profile are also presented in each part. Reprinted with permission from [1]. Copyright © 2017 Elsevier.

1.1.3 Laser Keyhole Welding

As one of the advanced joining technologies, laser keyhole welding has been energetically used in high precision joining of metal parts in the automotive, electronics, aerospace, shipbuilding, and medical industries [3]. Its major characteristics are compared with other popular welding techniques (*i.e.* arc and hybrid welding; hybrid: combination of laser and electrical arc) in **Table 1.1** [16].

Table 1.1. Comparison of arc, laser, and hybrid welding processes. Reprinted with permission from [16]. Copyright © 2009 Taylor & Francis.

	Arc welding	Laser welding	Hybrid welding
Gap bridging	Wide fusion zone Good gap bridging	Narrow fusion zone Poor gap bridging	Wide fusion zone Good gap bridging
Residual stress & Distortion	High	Low	Low
Productivity	Low welding speed Low productivity	High welding speed High productivity	Relatively high welding speed High productivity
Cracking propensity	High propensity for solidification cracking	Formation of brittle phases Increased propensity for cracking	Reduced amounts of residual stress Low propensity for cracking
Cooling rate	Low cooling rate Prevents brittle phase formation	High cooling rate Relatively high amounts of porosity	Relatively low cooling rate Reduced amounts of porosity
Weld penetration	Relatively shallow fusion zone Reduced penetration Multi-pass welding	High energy density Deep weld pool Single pass welding of thick sections	Relatively deep weld pool Single pass welding of thick sections
Stability	Reduced arc stability at higher welding speeds leads to humping	Keyhole instability	Interaction between laser and arc stabilizes the arc

In **Figure 1.6(a)**, power densities with corresponding bead shapes are presented according to the typical welding heat sources (laser, electron beam, plasma, and arc) [17]. As seen, laser and electron beam have a very high energy density compared with others, thus keyhole evolving during the welding is narrow and deep, which implies the narrow and deep weld beads. Therefore, using laser beam as a heat source, narrow and deep penetrated weld bead can be achieved with high strength, as shown in **Figure 1.6(b)** (a cross-sectional weld bead with the keyhole and heat affected zone (HAZ) indications) and **Figure 1.6(c)** (typical weld bead surfaces).

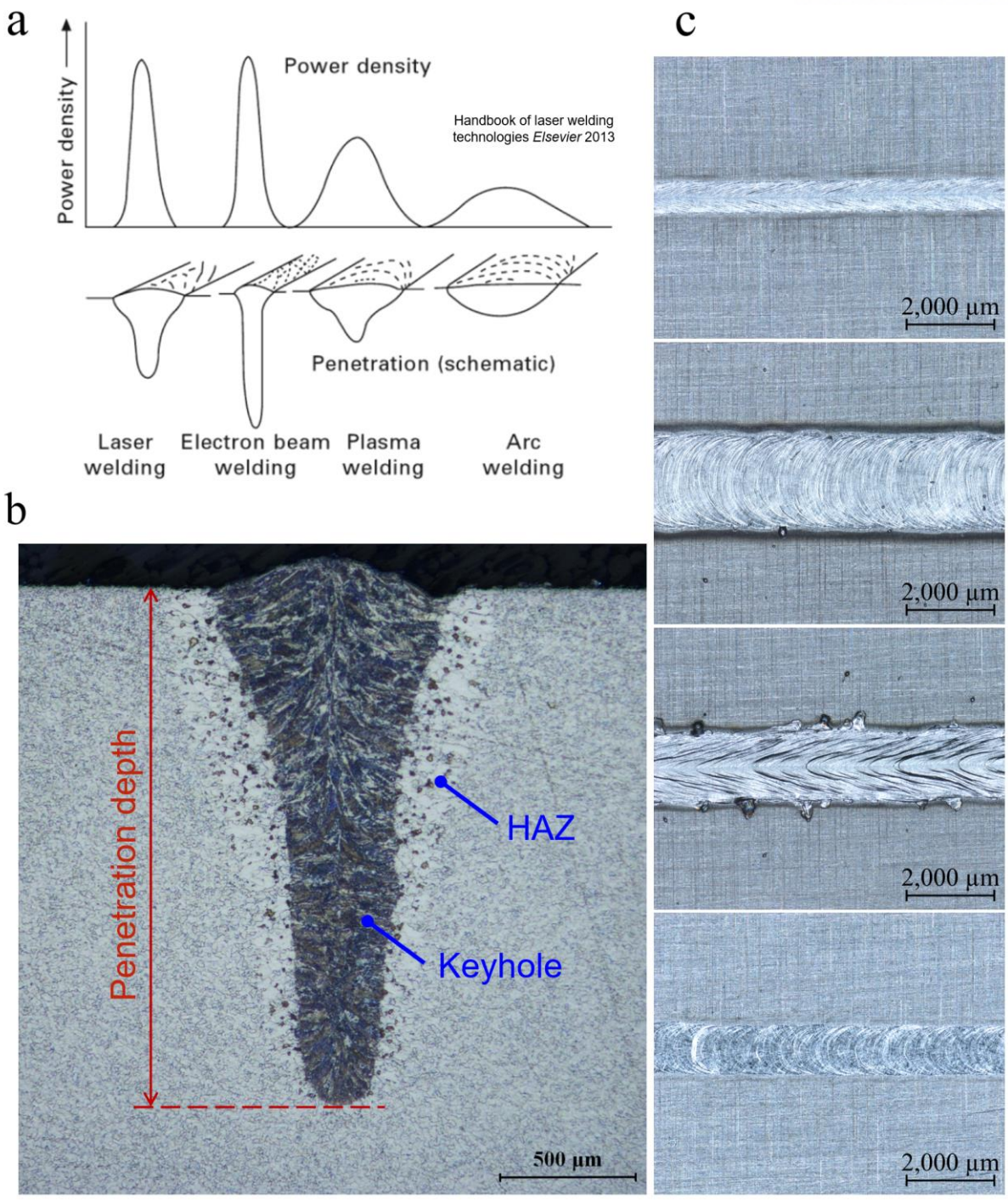

Figure 1.6. (a) Power densities and weld beads according to typical welding heat sources. Reprinted with permission from [17]. Copyright © 2013 Elsevier. (b) Cross-sectional laser weld bead with the keyhole and heat affected zone (HAZ) indications. (c) Various laser weld bead surfaces.

In **Figure 1.7(a)**, coaxially observed top and bottom weld surfaces during the process are presented [18], and a 3-D schematic of the laser keyhole welding is presented in **Figure 1.7(b)**. Since the laser beam has a very high energy density, when irradiated to the material, the material melts and evaporates, forming a vapor-filled capillary called the keyhole. Therefore, four phases of solid, liquid, vapor, and

plasma coexist in laser keyhole welding, involving heat conduction, melting, vaporization, plasma formation, and laser-plasma interaction [19, 20]. In addition, high interfacial forces such as capillary and thermocapillary (caused by large temperature gradient on liquid/vapor interface) forces and recoil pressure (caused by vapor flux) act at the same time, which makes the keyhole unstable and predictions of keyhole difficult. Also, multiple reflection effect of the laser beam comes into play as the keyhole deepens. In these reasons, there has been major difficulties in developing a reliable predictive model for the laser welding process, which is essential to attain a desired weld bead. In **Figure 1.8**, two simulation results from [20, 21] are presented.

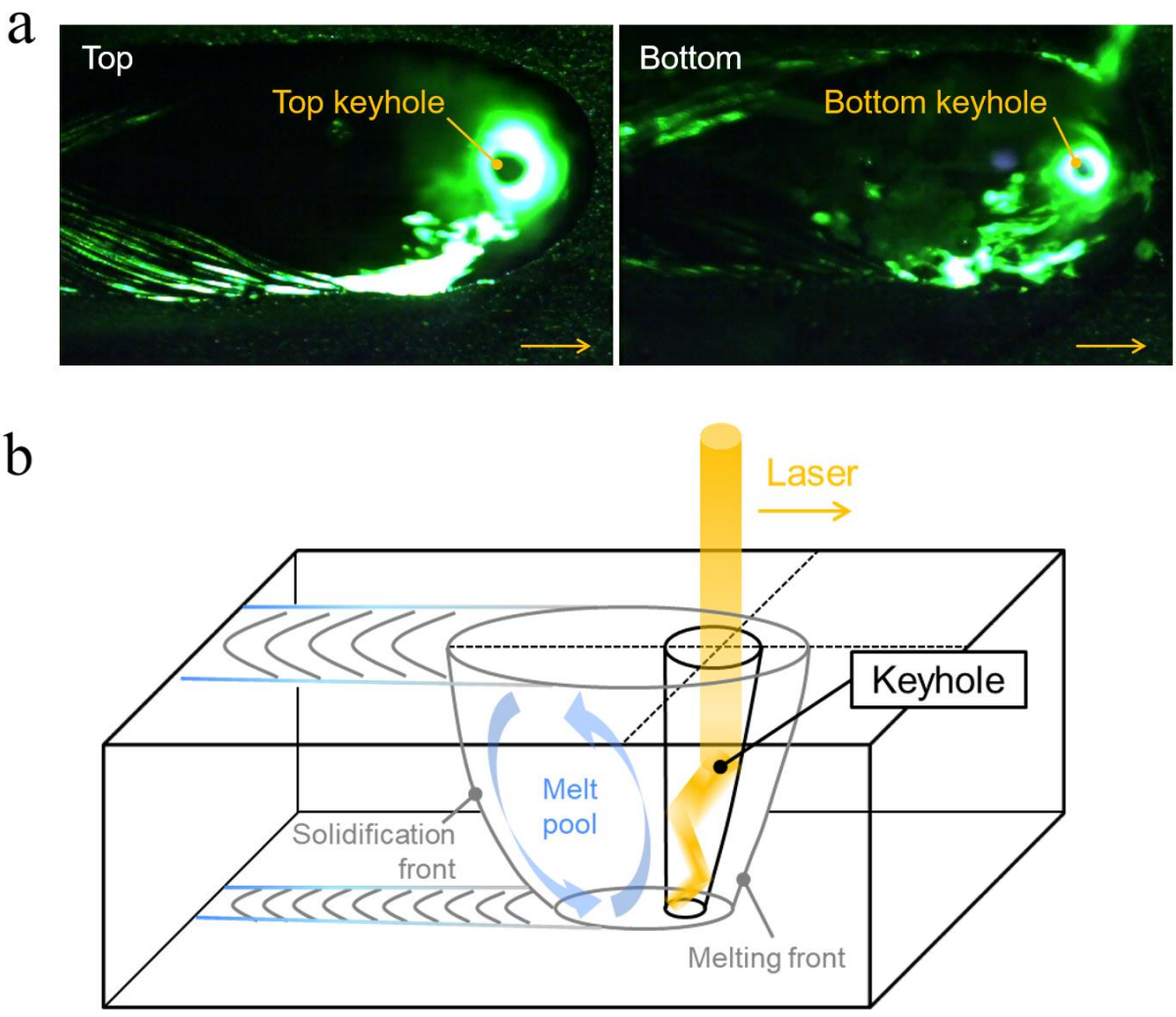

Figure 1.7. (a) High-speed camera images of top and bottom surfaces while laser welding. (b) Schematic of laser keyhole welding.

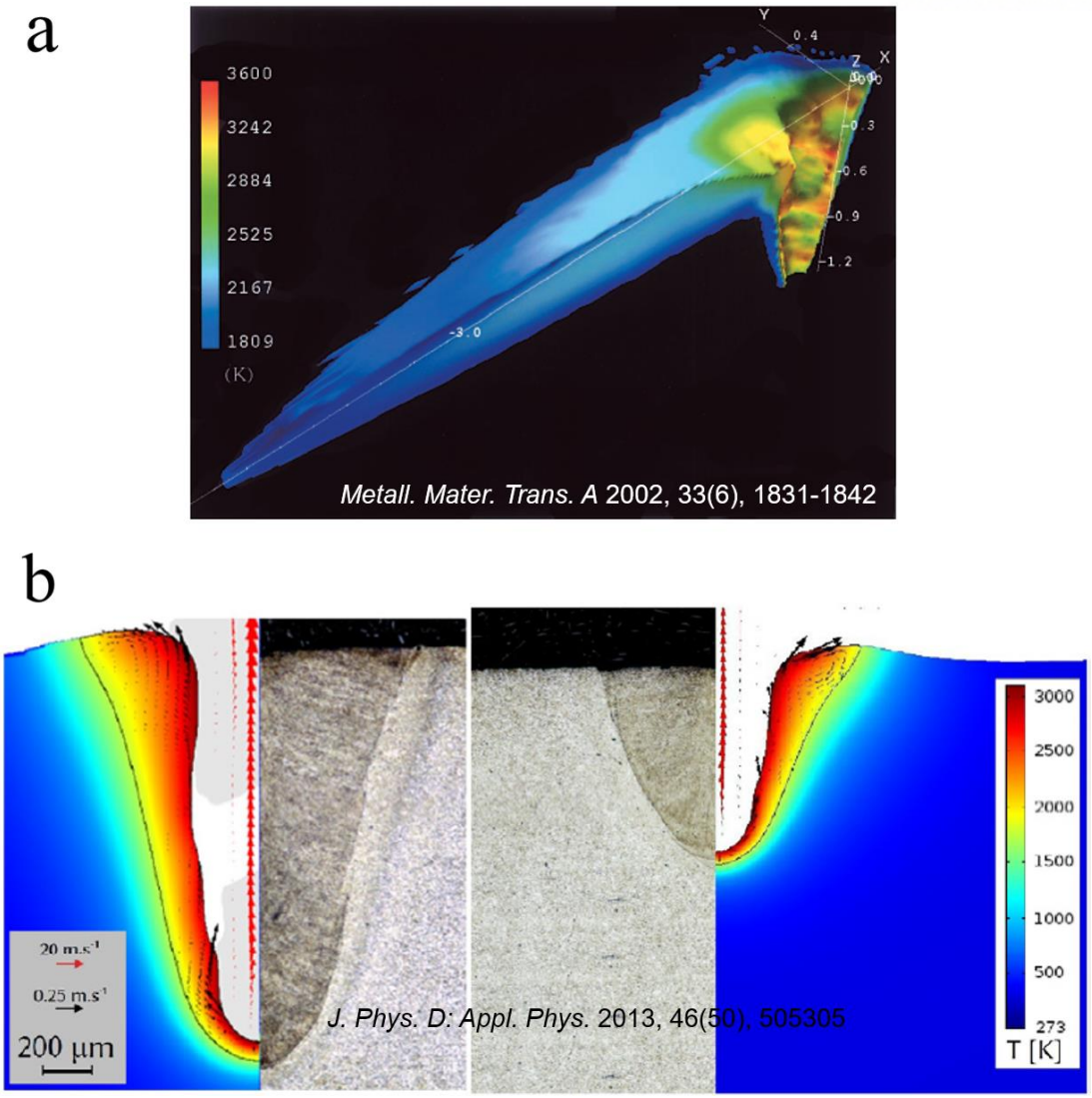

Figure 1.8. (a) Calculated melt pool shape with temperature distribution. (b) Comparison between the predicted melt pool shape and optical microscopic image. Reprinted with permission from [20, 21], respectively. Copyright © 2002 Springer Nature and Copyright © 2013 IOP Publishing Ltd, respectively.

While the developed physics-based predictive models have been useful in understanding the nature in it, however, most of them require very expensive computational cost, even simulating few seconds of the process (somewhat low applicability to the real industry). Also, according to the on-site welding conditions, the prediction accuracy is not fully guaranteed. In lieu of the traditional simulation, in this dissertation, we proposed a new deep learning-based framework to predict the high-resolution optical microscopic (OM) image of cross-sectional laser weld bead from two inputs of laser intensity and beam interaction time, in a few seconds with easy applicability (presented in **chapter 3**).

1.1.4 Self-Piercing Riveting (SPR)

Among several popular joining techniques such as welding, brazing, screws, bolted joints, clinching, nails, adhesives, riveting, and soldering, self-piercing riveting (SPR) is introduced in this section, which is a process of combining two or more sheets using a rivet without a preprocessed hole [22]. It can easily join the sheets regardless of the material types including steels, nonferrous metals, plastics, coated materials, and even carbon fiber composites, without heat, fumes, chips, and dusts [23, 24]. The SPR joint demonstrates high strength with good fatigue performance, and easy automation of the process (rapid cycle within 1~4 seconds, thus good reproducibility) [25, 26]. A schematic of the SPR process is presented in **Figure 1.9(a)** [27]: 1) clamping; 2) piercing upper sheet, pushed by a punch; 3) full piercing upper sheet and flaring of rivet legs in lower sheet; 4) releasing. In **Figure 1.9(b)**, two types of SPR joint are presented, which were carbon fiber reinforced plastic (CFRP) – galvanized steel (left) and steel alloy – aluminum alloy (right).

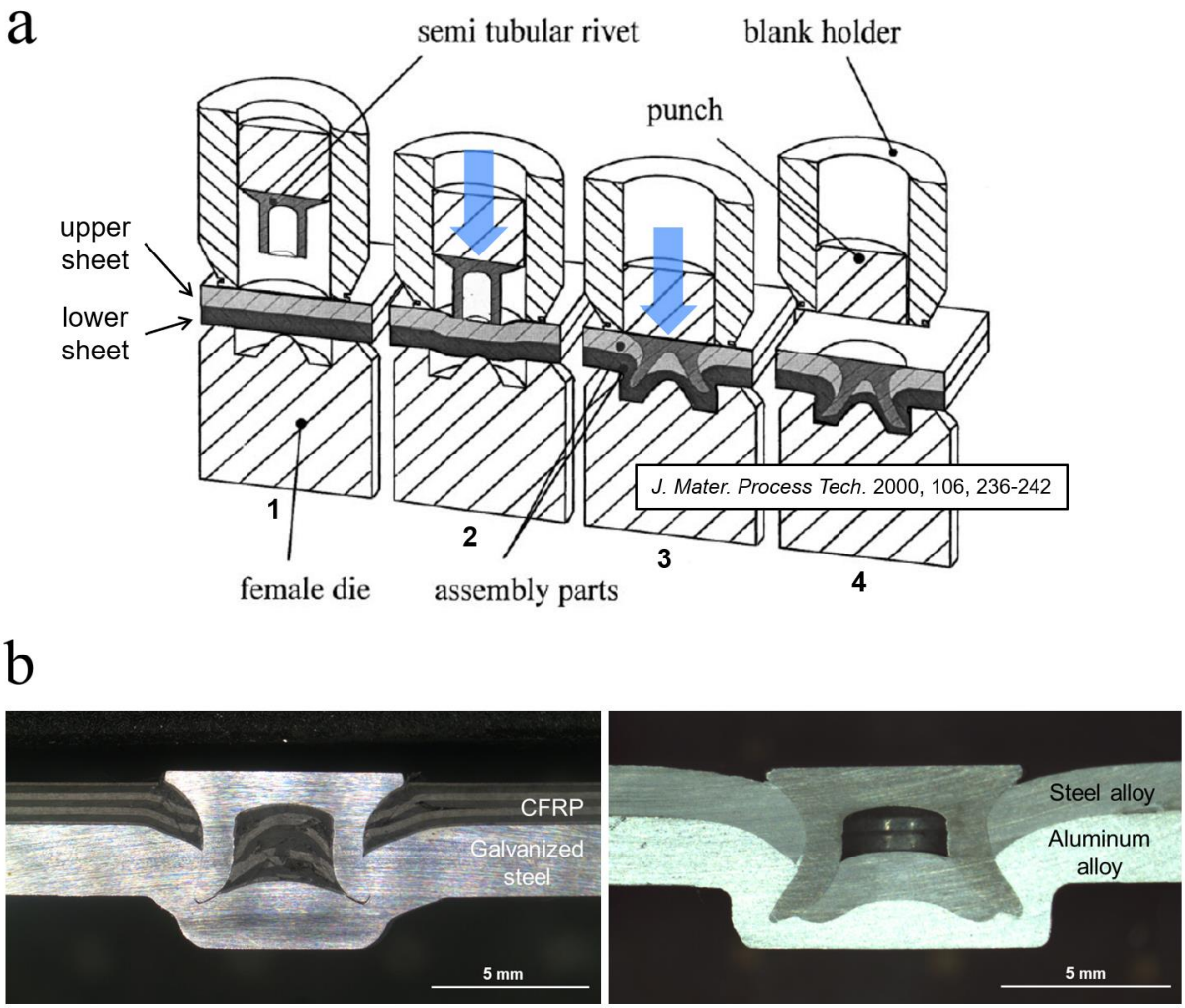

Figure 1.9. (a) SPR process. Reprinted with permission from [27]. Copyright © 2000 Elsevier. (b) SPR joints – CFRP/galvanized steel (left) and steel alloy/aluminum alloy (right).

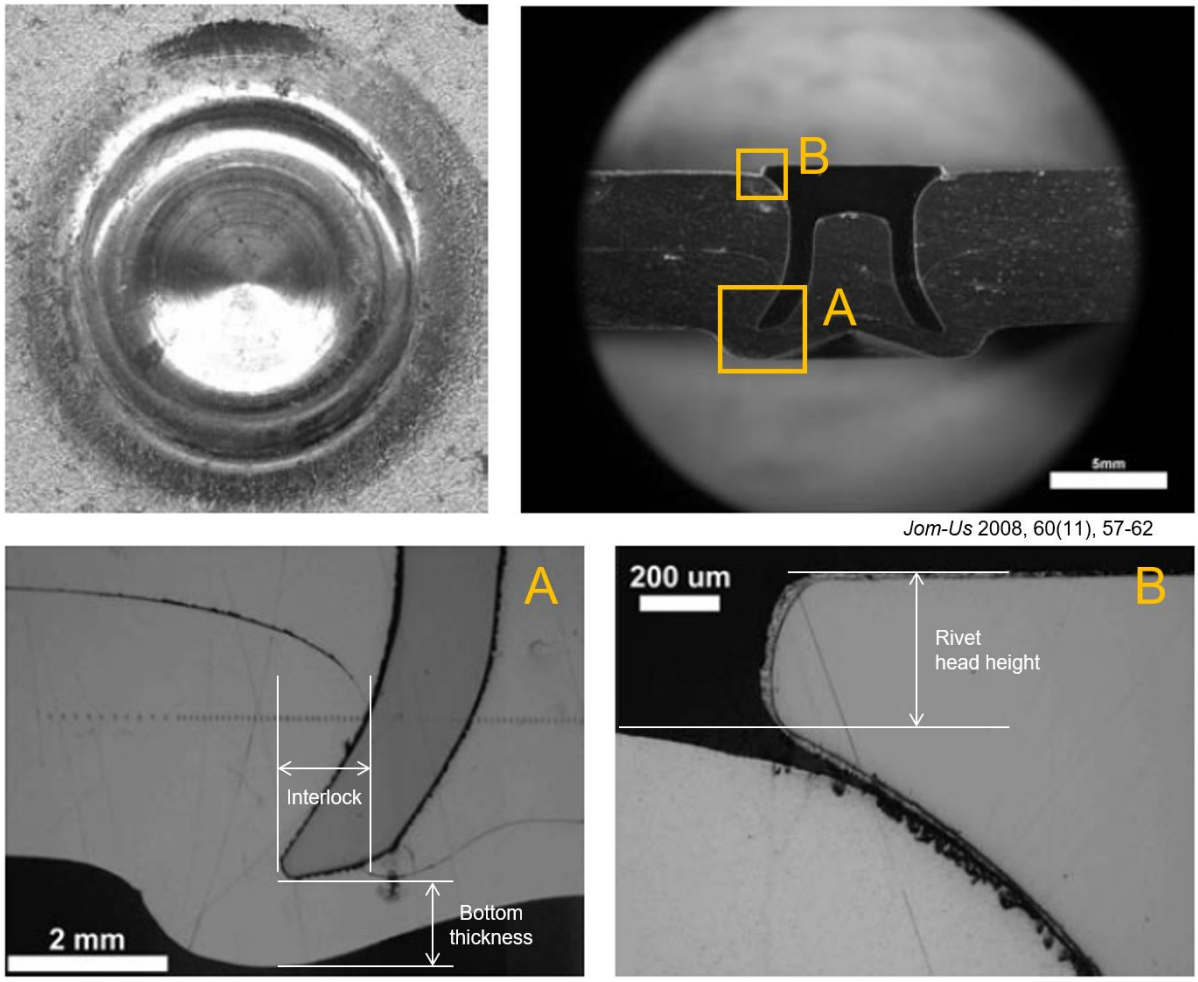

Figure 1.10. SPR joints observed from die and cross section. Reprinted with permission from [28]. Copyright © 2008 Springer Nature.

The SPR joint images observed from die and cross section are presented in **Figure 1.10** [28], with the magnified areas indicating interlock and bottom thickness (A) and rivet head height (B), which are the three main geometrical factors determining joint quality. (Good joint quality: small head height, moderately large interlock length, and thin bottom thickness) They are affected by various SPR process parameters, such as geometrical shape and dimension of rivet (diameter, length), sheets (thickness, edge distance), blank holder (diameter, pressure), and die (diameter, depth), material properties (hardness), punch force and speed [26]. Therefore, prediction of the cross-sectional shape from the process parameters has been a major concern in modeling the SPR process. Generally, finite element method (FEM) has been widely adopted to the prediction. Instead, in this dissertation, we propose a novel artificial intelligence (AI)-based cross-sectional shape predictive architecture, taking advantages of deep learning (**chapter 4**).

1.2 Introduction of Deep Learning

1.2.1 General Introduction of Artificial Intelligence and Deep Learning

At the time of writing this dissertation, the world has already entered the fourth industrial revolution, that is to say, an all-in-one superintelligence revolution by cyber-physical system (CPS), in which artificial intelligence, internet of things (IoT), big data, and cloud computing systems are integrated in real life [29]. One of its cores is the various splendid deep learning algorithms, such as convolutional neural network (CNN), recurrent neural network (RNN), reinforcement learning (RL), generative adversarial network (GAN), etc. A diagram for the deep learning, machine learning, and AI is presented in **Figure 1.11**, with their definitions [30-32]. In this dissertation, the focus was primarily placed on the computer vision area in deep learning, which mainly uses CNN (for image recognition; **chapter 1.2.2**) and GAN (for image generation; **chapter 1.2.3**) structures.

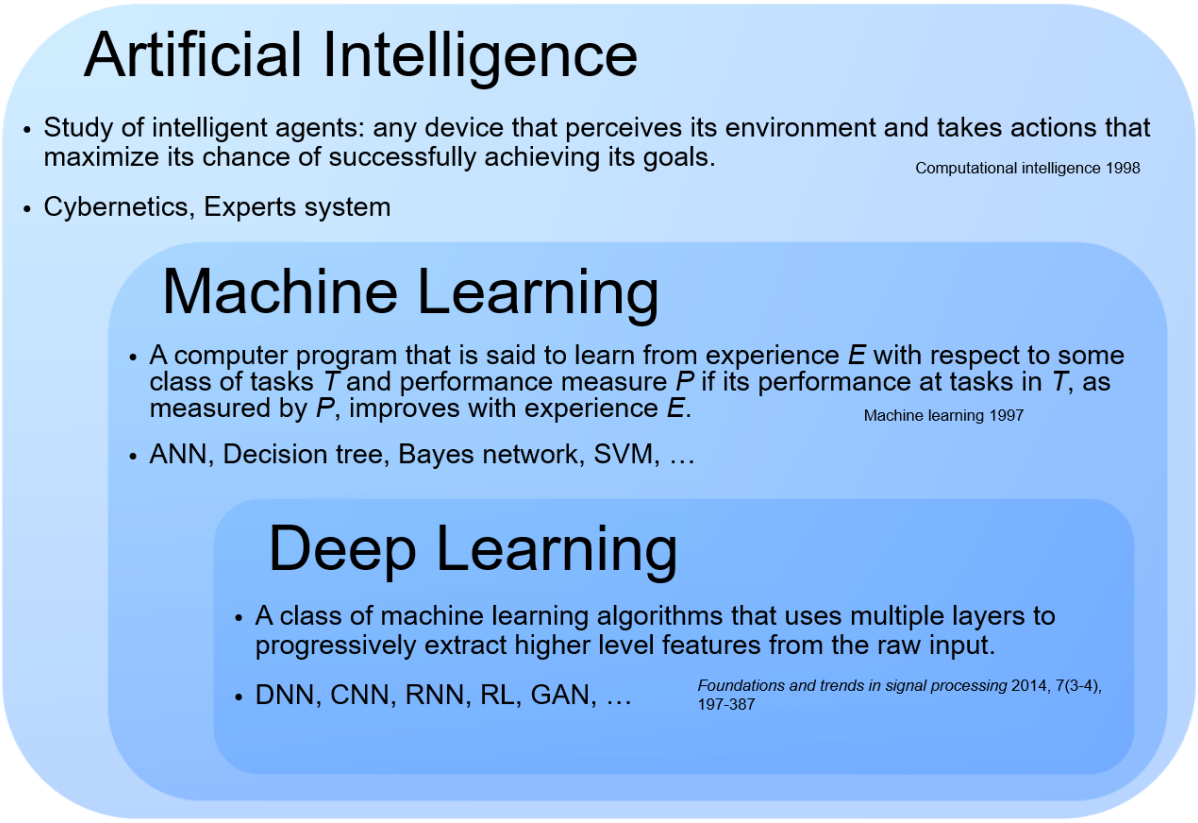

Figure 1.11. Diagram for artificial intelligence, machine learning, and deep learning ([30-32], respectively).

1.2.2 Image Recognition

For recent years, recognizing unique intrinsic patterns and information contained in images using AI has been a hot trend. Specifically, there are three major fields in image recognition, namely, image classification, object detection, and segmentation (**Figure 1.12**), which are the tasks of classifying label of the image, localizing object as well as classifying its label, and pixel by pixel classification, respectively [33].

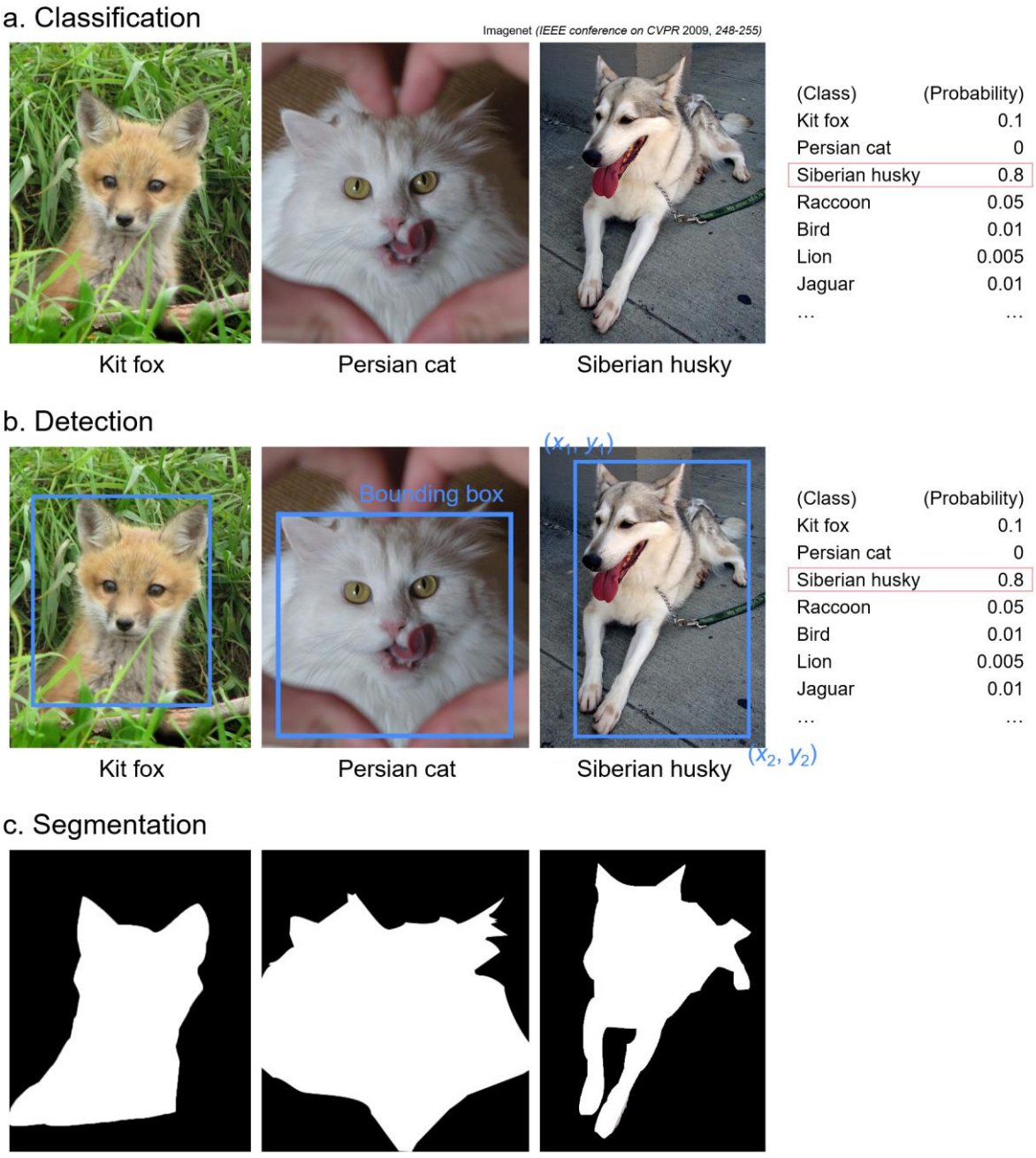

Figure 1.12. Three image recognition methods – classification, detection, and segmentation. The presented kit fox, Persian cat, and Siberian husky images were taken from the Imagenet [34].

History for the state-of-the-art deep learning models on each task (*i.e.* image classification, object detection, semantic segmentation, and instance segmentation) is presented in **Figure 1.13**. As shown, entering mid-2010s, the computer vision is at its heyday, owing to an explosive growth in various deep learning algorithms as well as high-performance GPU which enables stacking deep convolutional layers.

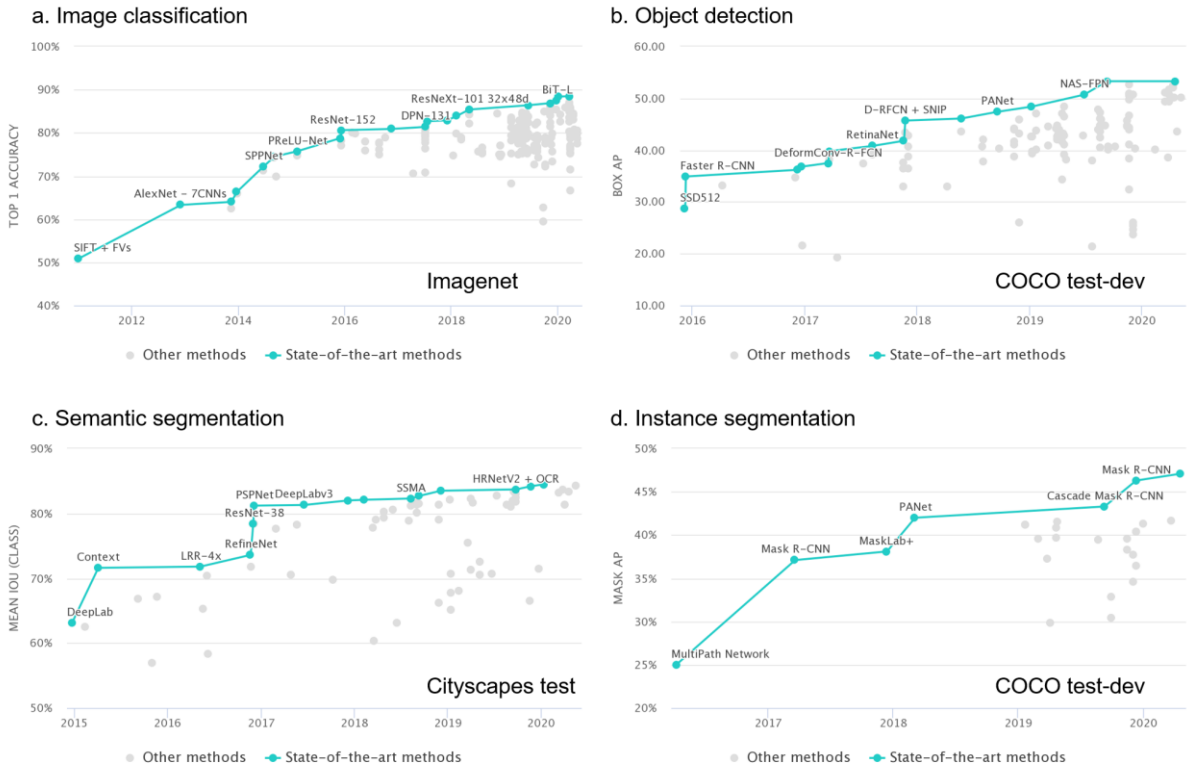

Figure 1.13. History of the state-of-the-art AI models on each image recognition task. All graphs (https://paperswithcode.com; accessed at May 2020) are utilized under a Creative Commons license (CC BY-SA 4.0).

The convolutional layer [35] is used in every image recognition model, and its schematic diagram is presented in **Figure 1.14**. As shown, the convolution is an operation of scanning input image tensor (three-dimensional tensor; x; shape: [height (H), width (W), in-channels (C)]) using a trainable filter weight tensor (four-dimensional tensor; w; shape: [filter height (FH), filter width (FW), in-channels (C), filter number or out-channels (FN)]). In the figure, following calculations are conducted with each filter:

$$\begin{aligned} y_{11} &= \sum_{i=1}^{3}\sum_{j=1}^{3} x_{ij} \cdot w_{ij} + b \\ y_{12} &= \sum_{i=1}^{3}\sum_{j=1}^{3} x_{i,j+2} \cdot w_{ij} + b \\ &\vdots \end{aligned} \quad . \tag{1.4}$$

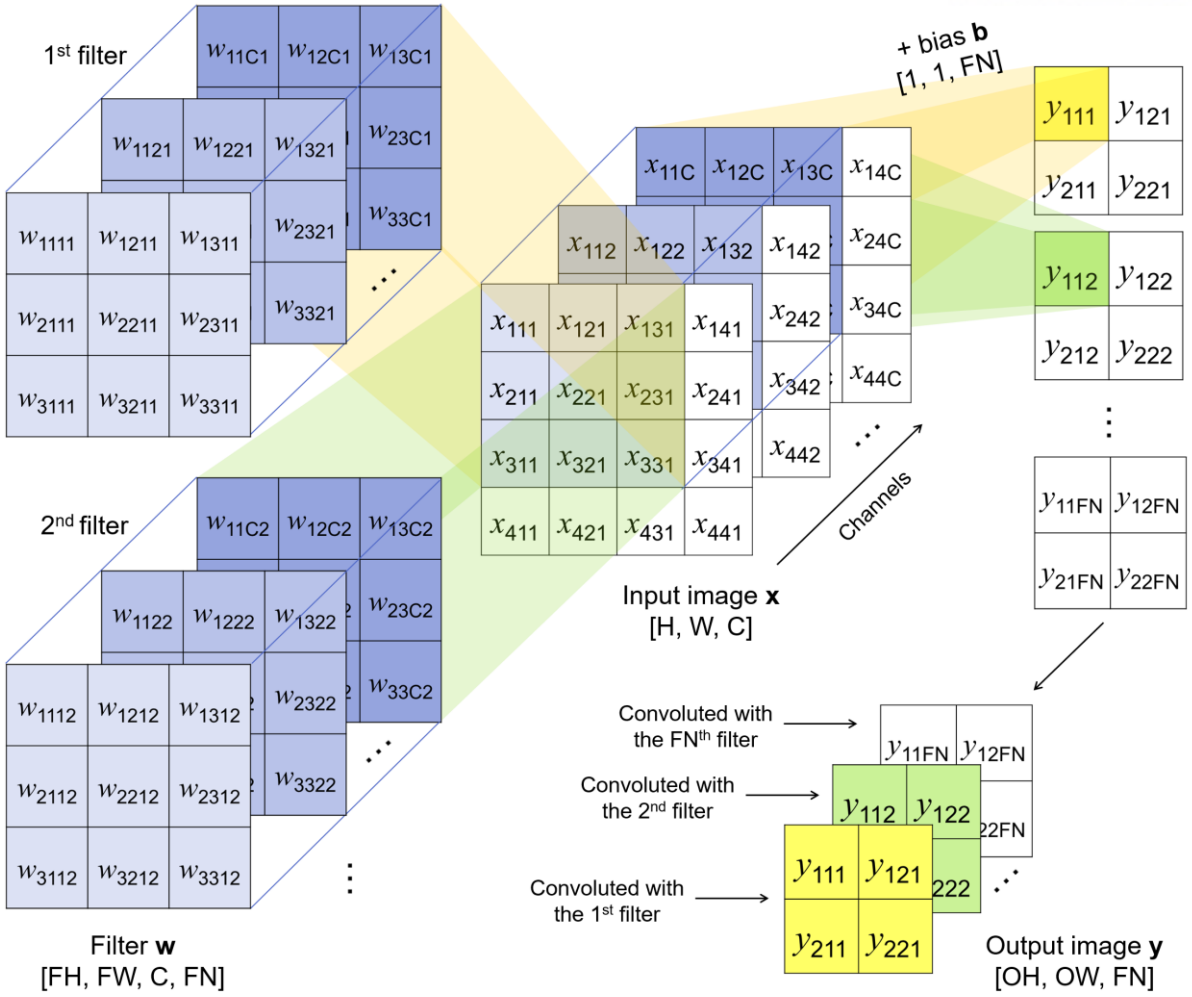

Figure 1.14. Schematic diagram for the convolutional layer.

1.2.3 Image Generation

Generation of images with specific objectives or containing unique patterns has been a boom after the emergence of GAN [36]. Combining the CNN and GAN structures [37], supervised image translations, unsupervised image translations, super-resolutions, high-resolution image translations, and video translations have become possible (**Figure 1.15** and **Figure 1.16**), and actively employed in various areas such as medical imaging and embedded systems (smartphone, car camera, CCTV, etc.).

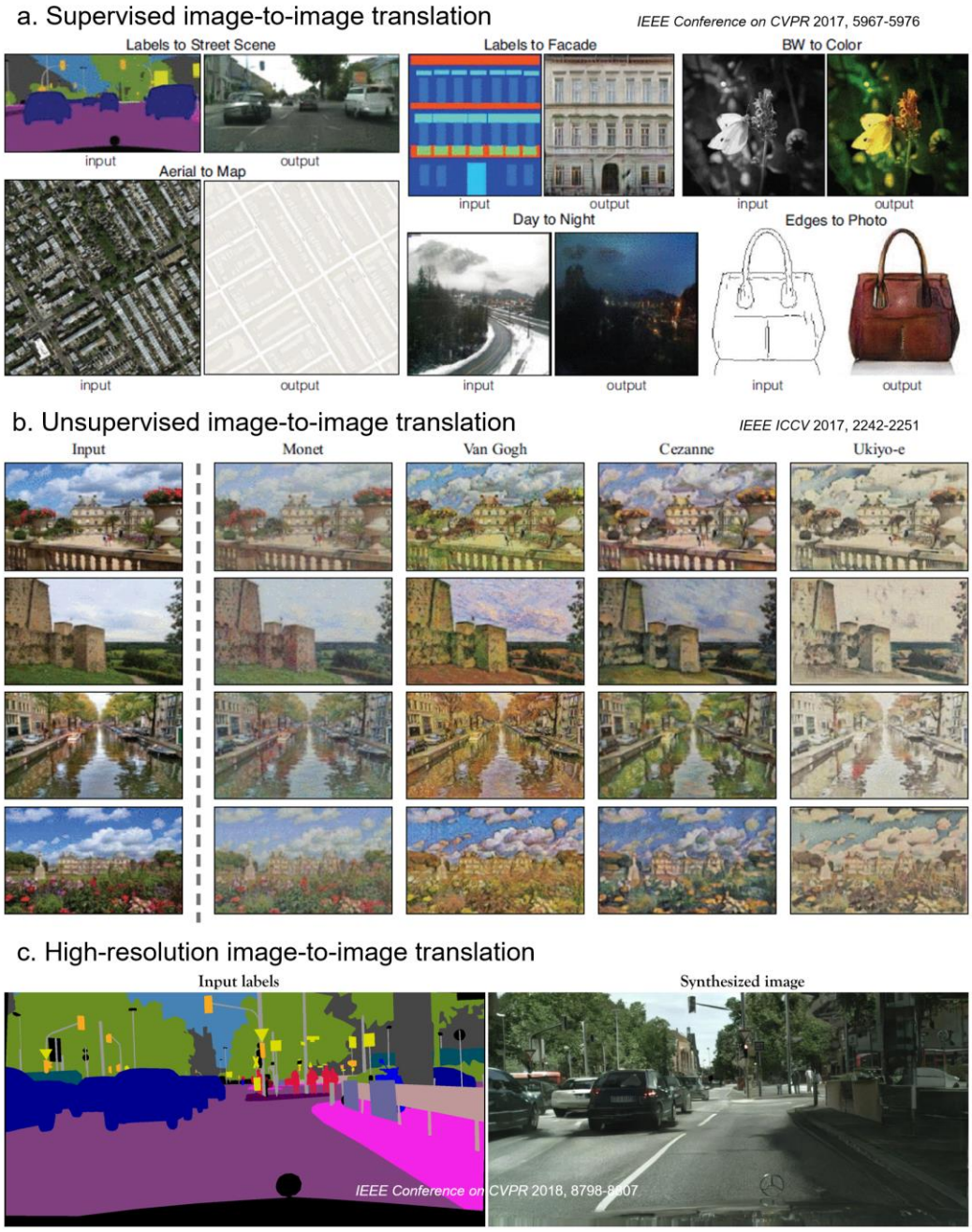

Figure 1.15. Representative images for supervised, unsupervised, and high-resolution image-to-image translations. Reprinted with permission from [38-40], respectively. Copyright © 2017 IEEE, Copyright © 2017 IEEE, and Copyright © 2018 IEEE, respectively.

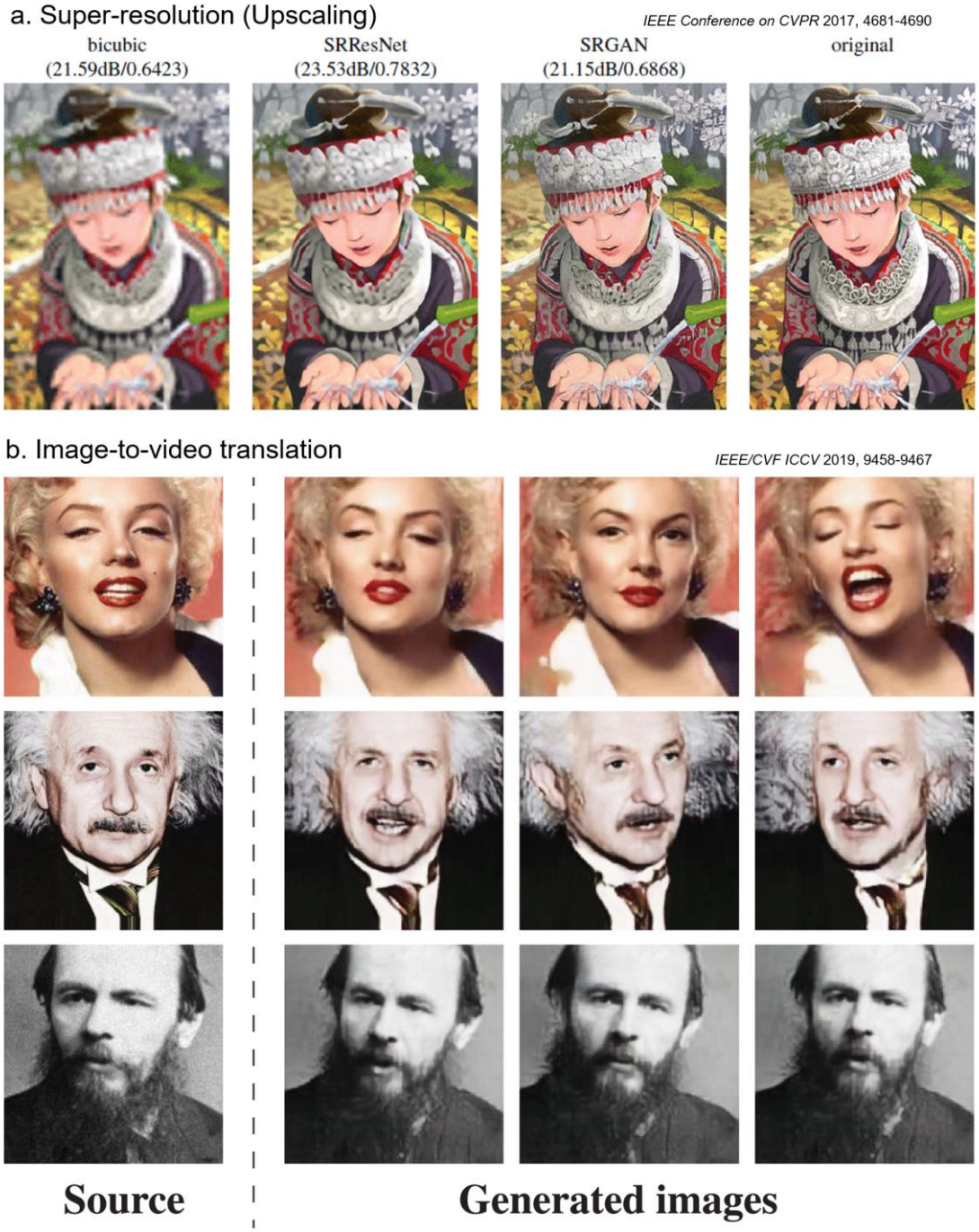

Figure 1.16. Representative images for super-resolution (upscaling) and image-to-video translation. Reprinted with permission from [41, 42], respectively. Copyright © 2017 IEEE and Copyright © 2019 IEEE, respectively.

Also, the CNN plus GAN structure can be successfully utilized in modeling of materials processing technologies, by replacing the data with experimental or simulation results. Trained from the given data, the CNN and GAN-based deep learning model can produce predictive images from the recognized features and patterns (thus, data-driven). All presented deep learning models in this dissertation were

based on this prediction mechanism.

Specifically, in laser heat treatment, hardness predictive images were generated from the input of cross-sectional temperature distribution. Unlike the traditional physics-based prediction model which required a complete temperature history during the entire heat-treating process, the presented deep learning model only required the one temperature distribution when surface temperature was the peak. The deep learning model spontaneously recognized the thermal characteristics contained in the two-dimensional cross-sectional temperature distribution, and translated it to the hardness distribution via deep convolutional layers. (Information about the heating as well as cooling processes are already included in the cross-sectional temperature distribution.) The translated hardness distributions matched well with the experimental results.

In laser keyhole welding, high-resolution weld bead predictive images were generated from the input of laser processing parameters (laser intensity and beam interaction time). Unlike the traditional physics-based method which required solving governing equations for heat conduction, melting, vaporization, plasma formation, laser-plasma interaction, and multiple reflection, the proposed deep learning model only needed the laser process conditions. The predicting line was divided into two, such that the weld bead segmentation map was first predicted and then it was synthesized to the OM image through the following line, since the amount of information contained in the two parameters was limited to construct the high-resolution OM image. The AI model well recognized the necessary information contained in the inputs (laser processing parameters and weld bead segmentation map) and successfully generated the weld bead image.

In SPR process, cross-sectional shape predictive images were created from the input of punch force. Instead of the conventional finite element simulation, data-driven deep learning model was adopted, which was trained from the experimentally-obtained data (OM image). To characterize the shape of materials, concept of image segmentation was first employed, which was also utilized using a deep learning model. In the segmentation model, SPR OM images for two sheet combinations were successfully segmented at a human level, and used for training of the cross-sectional shape predictive model. The cross-sectional shapes were well predicted in terms of the foremost geometrical quality factors, which implies that the model spontaneously understood the mapping between the input punch force and cross-sectional shape and was trained at a semantic level.

Chapter 2. Deep Learning Study on Laser Heat Treatment

This chapter includes the published contents:

2.1 Introduction

Laser heat treatment has been widely employed for various industrial applications, as a method for enhancing the strength, hardness, and wear resistance of steel parts. A laser beam is a high-intensity heat source with which precise, localized hardening is possible, and owing to its fast cooling rate, the hardening quality is generally superior to those of other heat treatment methods [1, 7]. Laser heat treatment is a complex multiphysics process, where the optical energy of a laser beam is converted to heat at the surface of the laser-irradiated part, and a desired hardness distribution is produced as the heat flows through the part.

To obtain the desired hardness distribution, however, a method for accurately predicting the outcomes of the process is necessary, and a great deal of effort has been made by various researchers toward this purpose [9-12, 43-47]. Because laser heat treatment is a thermal process, most of the prediction methods have been based on thermal analyses. Kou et al. [43] conducted a 3-D thermal simulation of laser transformation hardening using the finite difference method, and experimentally investigated the hardened zone. The model predicted thermal histories, but was not capable of predicting hardness distributions. Meijer and Sprang [44] showed that, with a constant hardening depth, the hardening results can be optimized with respect to only one independent parameter, the maximum surface temperature. Komanduri and Hou [45] obtained an analytical solution for the temperature field in laser transformation hardening of steel based on the moving heat source solutions. Although this solution is simple and helpful, its prediction capacity is somewhat limited owing to its analytical nature. Leung et al. [46] derived an exact solution for the laser transformation hardening process, and the solution was illustrated for a customized laser beam with a flat-top rectangular configuration. Using the solution, they predicted the hardness and hardening depth, which were in good agreement with the experimental results. Martínez et al. [47] developed a closed loop temperature control for the laser hardening process with scanning optics (LTHS), and employed thermal simulations to minimize experimental work. Ki and So [9] proposed a method for analyzing the laser hardening process by constructing process maps for effective carbon diffusion time and effective cooling time. Using the

process map approach, So and Ki [10] studied how the laser hardening process changes as the specimen thickness is varied, and Ki et al. [11, 12] investigated the laser hardening of steel sheets by employing a heat sink, where they studied the effects of thermal contact resistance and heat sink thermal conductivity on the hardenability. Although the process map approach was useful in understanding the hardening process on a large process window, due to the 1-D model adopted, the prediction accuracy was found to deteriorate at high interaction times. Recently, Oh and Ki [1] improved the accuracy by employing a 3-D thermal simulation and proposed predictive models for hardness distributions and deflection angles.

As presented, the majority of the previous models are physics-based. Instead, in this dissertation, we proposed a deep learning-based data-driven modeling of the process, which can make predictions by learning from the experimental or simulation data. Although deep learning is becoming widely employed in many engineering fields, very little research has been done in the laser material processing field. Recently, Günther et al. [48] developed a self-learning and self-improving laser welding system using deep neural networks and a reinforcement learning technique, which enabled the laser to deliver situation-appropriate powers. Several studies using a simple artificial neural network (ANN) have been reported for laser heat treatment [49-51]; however, no deep learning-based model has been reported yet.

In this dissertation, for the first time, a deep learning based predictive model has been developed for laser heat treatment of AISI H13 tool steel, where a conditional generative adversarial network (cGAN) architecture with a CNN-based encoder-decoder [38, 52] is used to predict hardness distributions. The cGAN architecture [36, 52] is known to show superior performance in image-to-image translations, and in this study, a cGAN architecture was employed for temperature-to-hardness mappings with a CNN structure [35].

In this dissertation, a 2-D temperature distribution on the cross-section of a specimen (which was computed from a 3-D thermal analysis and obtained right at the moment when the surface temperature reached the maximum on the given cross-section) was provided as an input to the model, and the experimentally measured hardness distribution was also supplied as the answer to the input temperature distribution. Then, the network was trained to translate the input temperature distribution into a hardness distribution. When compared with the authors' previous model based on the effective carbon diffusion time[1], the developed model improved prediction accuracy to 94.4% from 84.8%.

2.2 Data Preparation

Deep learning is a data-driven modeling approach, and it is important to prepare good data sets that can produce accurate results effectively. **Figure 2.1** shows how computational and experimental data were utilized for the model developed in this study. First, a temperature profile on the cross-section of a steel specimen was calculated using a 3-D heat conduction model. For each process condition, the temperature distribution at the moment when the surface temperature reaches the maximum was chosen, *i.e.*, when the laser beam just passes by the given location. The details of the simulation are given in **chapter 2.2.1**. Second, experiment of laser hardening was conducted using the same process conditions as the ones used for numerical simulations. For each heat treated specimen, micro Vickers hardness was measured for the entire cross-section. The experimental details are explained in **chapter 2.2.2**. In **chapter 2.2.3**, the data preprocessing procedure is described in detail.

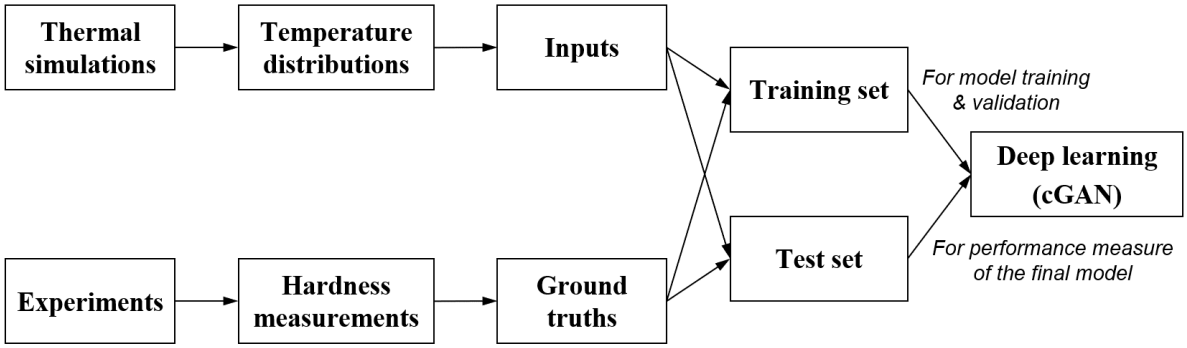

Figure 2.1. Both computational and experimental data were used for training and testing the deep learning model.

2.2.1 For deep learning inputs: 3-D thermal simulation

Figure 2.2(a) shows a schematic drawing for the laser heat treating of an AISI H13 tool steel specimen. To obtain temperature distributions inside the specimen, which is the input to our deep learning model, a 3-D heat conduction equation was numerically solved using the finite element method:

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right), \tag{2.1}$$

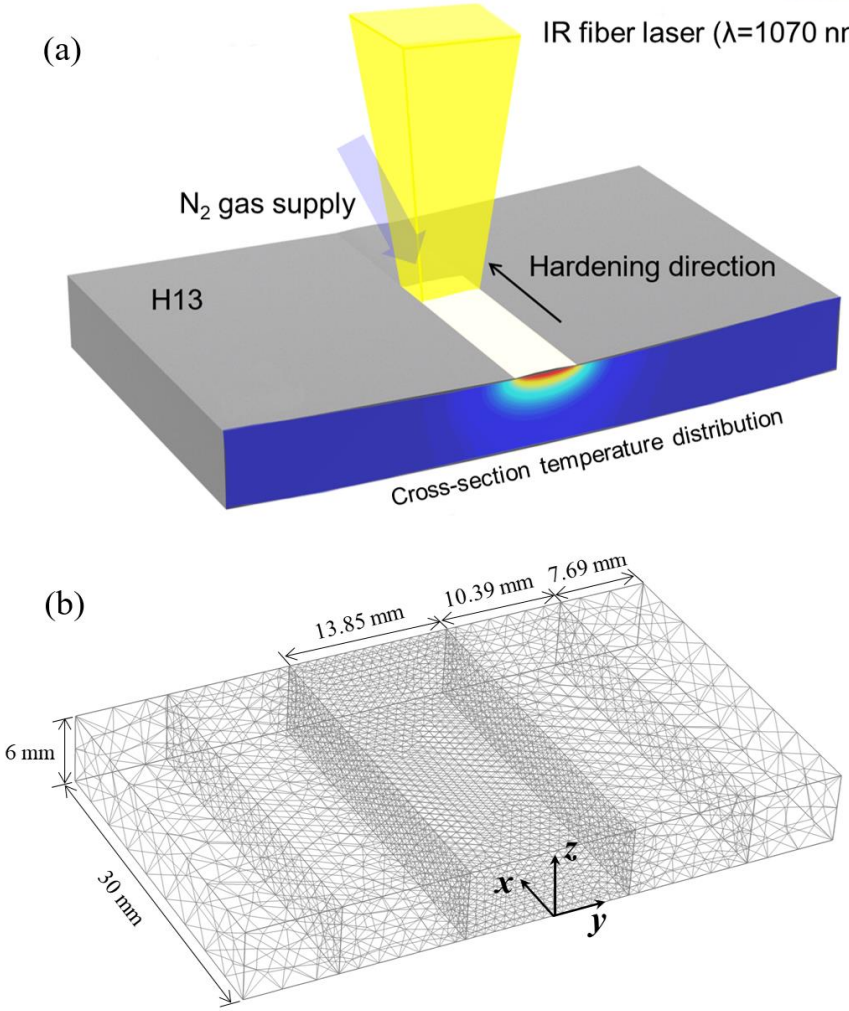

Figure 2.2. (a) Schematic of the laser heat treatment process. (b) Computational domain shown with dimensions, mesh used, and the coordinate system.

where T, t, and α denote temperature, time, and thermal diffusivity, respectively (x, y, and z are spatial coordinates). The computational domain has the dimensions 30 mm × 50 mm × 6 mm, and information on the mesh used and the coordinate system is given in **Figure 2.2(b)**. Note that, from the authors' previous study [1], the specimen thickness of 6 mm was found to be thick enough to fully reflect the characteristics of very thick plates (with the maximum self-quenching effect) for the process window considered in this study. The laser beam (continuous-wave, 1070 nm) was modeled as a rectangular moving heat source with a uniform intensity distribution. It was applied at the top surface along the x-direction and can be mathematically written as

$$I(x,y,z=L_z,t)=\begin{cases}\dfrac{\eta P}{l_x l_y} & \text{if} \quad -l_x/2 \le x-Vt \le l_x/2 \ \text{ and } \ -l_y/2 \le y \le l_y/2 \\ 0 & \text{if} \quad \text{otherwise}\end{cases}, \quad (2.2)$$

where I is the laser intensity distribution, L_z is the domain thickness (6 mm), l_x and l_y are laser beam lengths in the x- and y-directions, respectively, and η, P, and V are laser beam absorptivity, laser power, and laser scanning speed, respectively. As shown in **Figure 2.2(a)** and **Equation 2.2**, the laser beam moves in the +x-direction and heats the specimen from end to end. In this study, temperature-dependent material properties of AISI H13 steel (thermal diffusivity, density, and specific heat) taken from the COMSOL material library ('H13 (UNS T20813) (G 4404 SKD61) [solid]') were employed as shown in **Table 2.1**. The material properties and laser parameters used in the simulation are summarized in **Table 2.1**.

Table 2.1. Material properties and laser parameters used in the simulation, and the Vickers hardness of AISI H13 steel.

PROPERTY	VALUE
Thermal conductivity $k(T)$	$31.11 - 0.071T + 2.47\times10^{-4}T^2 - 2.95\times10^{-7}T^3 + 1.16\times10^{-10}T^4$ (W/m K)
Density $\rho(T)$	$7777.65 + 0.048T - 5.63\times10^{-4}T^2 + 2.70\times10^{-7}T^3$ (kg/m^3)
Specific heat capacity $c_p(T)$	$488.04 - 0.20T + 3.66\times10^{-4}T^2$ (J/kg K)
Laser beam size ($l_x \times l_y$)	3.7 × 4.4 mm^2
Beam absorptivity	0.4 (shielding) [53]
Surface emissivity ε on polished (top) and unpolished (elsewhere) surfaces	0.07, 0.79 [54]
Base hardness of AISI H13	200 Hv

The radiation and natural convection were considered as the boundary conditions [1]. For the top surface, for example, the boundary condition is written as

$$-k\left.\frac{\partial T}{\partial z}\right|_{z=L_z} = I(x,y,z=L_z,t) + h\left(T_\infty - T\right) + \sigma\varepsilon\left(T_\infty^4 - T^4\right), \tag{2.3}$$

where k, h, T_∞, σ, and ε are thermal conductivity, convection heat transfer coefficient, surrounding temperature (25 °C), Stefan-Boltzmann constant (5.670367×10^{-8} W/m^2 K^4), and surface emissivity, respectively. As an initial condition, the specimen was assumed to be at room temperature (25 °C)

before laser irradiation. Details on the boundary conditions including the heat transfer coefficient can be found in the authors' previous work [1].

To solve the heat conduction equation with the boundary conditions, the COMSOL Multiphysics® software was employed. A total of 85510 tetrahedral elements were used, and the computational domain was divided into three regions in the transverse direction (**Figure 2.2(b)**) and different element sizes were used depending on the distance from the centerline. The maximum element size along the laser path where the hardening process occurs was the finest at 19% of the transverse beam size (l_y, 4.4 mm), but it increased to 41.8% and to 92% as the distance the laser beam increased.

Figure 2.3 shows the eight process conditions (labeled as a, b, …, h) simulated using the afore-mentioned model on an intensity-interaction time (I-t_i) diagram. Out of eight conditions, three conditions (shown with red squares, c, f, and h) were used as a training data set and five (marked with blue squares, a, b, d, e, and g) as the test data set. (The validation was conducted by k-fold cross-validation, so validation set was not additionally assigned.)

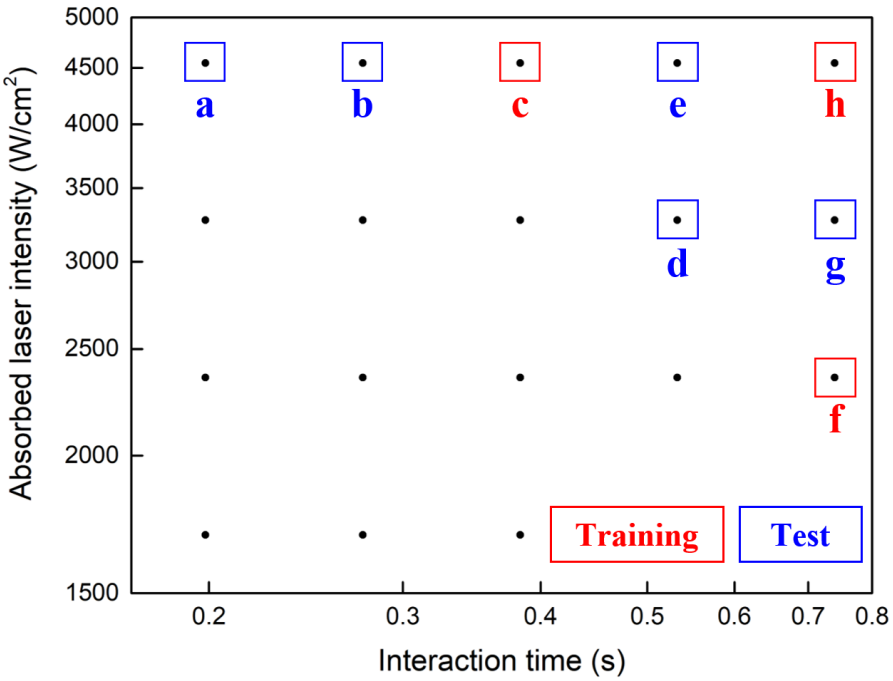

Figure 2.3. Eight process conditions considered in this study. They are shown on an intensity-interaction time diagram ($I=\eta P/(l_x l_y)$ and $t_i=l_x/v$). Process conditions shown in red and blue squares were used as a training data set and a test data set, respectively.

2.2.2 For ground truths: experiment

In this study, a 2 kW multi-mode fiber laser (IPG YLS-2000) with a wavelength of 1070 nm was used to heat-treat the AISI H13 steel specimen (**Figure 2.4**). The circular top-hat intensity profile of the laser beam was transformed into a rectangular top-hat intensity profile by using a beam homogenizer and focused on the surface of the specimen by a 250 mm focusing lens. The laser beam size was 3.7 mm (l_x) × 4.4 mm (l_y) (to be consistent with the simulation). The specimen was an AISI H13 tool steel block with a size of 30 mm × 50 mm × 6 mm (the same dimensions with the simulation; its chemical composition is given in **Table 2.2**).

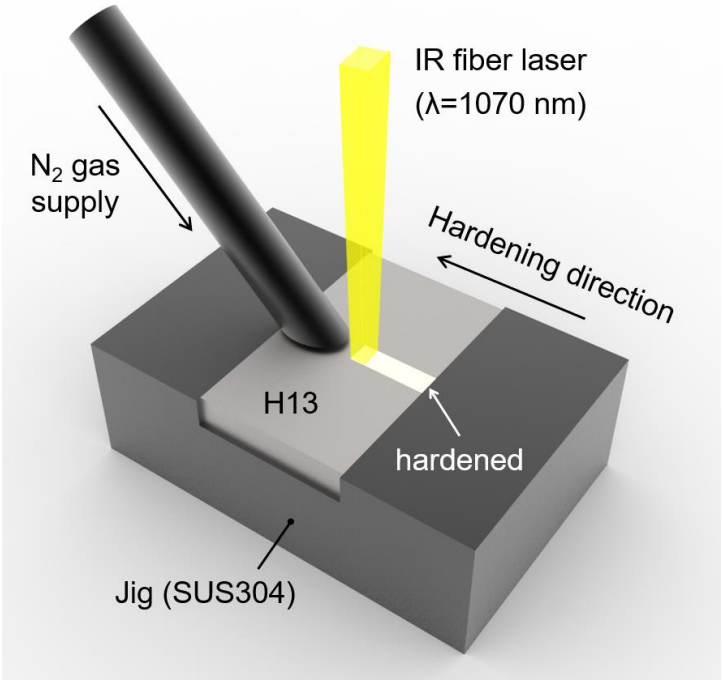

Figure 2.4. Schematic of the laser heat treatment process.

Table 2.2. Chemical composition of AISI H13 tool steel.

AISI H13	C	Si	Mn	Cr	Mo	V	Fe
(%)	0.384	0.981	0.388	5.113	1.038	0.820	balance

Experiments were conducted using the eight experimental conditions shown in **Figure 2.3**, and to prevent surface oxidation, N_2 gas was supplied on the specimen surface at a flow rate of 10 L/min. After the experiment, as described in **Figure 2.5**, micro Vickers hardness was measured for the cross-sections of all specimens by applying a load of 0.5 kgf for 15 s. As shown in **Figure 2.5(a)**, measurement

locations were selected around the HAZ horizontally from –25 mm to 25 mm (along the *y*-direction) with a separation distance of 250 μm and vertically from 5.9 mm to 0 mm (along the *z*-direction) with a separation distance of 100 μm, therefore covering the entire hardened region. The OM image of indentations inside the yellow box in **Figure 2.5(a)** is given in **Figure 2.5(b)**. In **Figure 2.5(c)**, an example of the indentation is presented (238 Hv).

To be used as ground truths for the model and compared with the predicted hardness distributions, the hardness values measured at discrete locations were interpolated using the bilinear interpolation technique, and a continuous distribution of hardness was obtained. As the separation distances between the indentation points were small (100–250 μm), interpolation results were reasonably accurate. From this interpolated measurement data, a ground truth image with the same size as both the temperature and predicted hardness distributions was obtained (296 × 301 pixels in size).

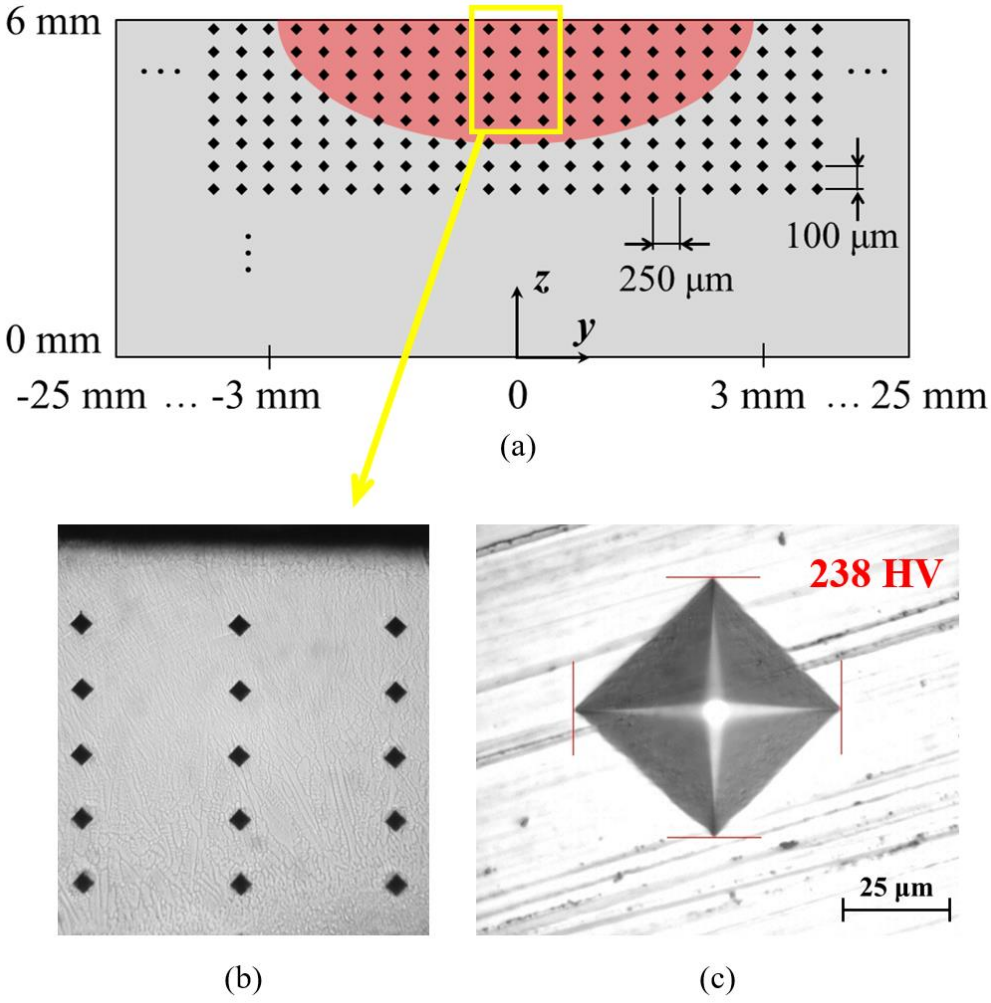

Figure 2.5. Hardness measurements on *y*-*z* plane, at *x*=15 mm. (a) Indenting domain; locations and separation distances. Red area denotes the HAZ. (b) OM image of indentations inside the yellow box in (a). (c) An example of the indentation. The hardness was measured by 238 Hv.

2.2.3 Training data set

In **Figure 2.6**, the three results (cases c, f, and h) in the training data set comprising of FEM-simulated temperature distributions (left figures) and measured hardness distributions (right figures) are presented side by side.

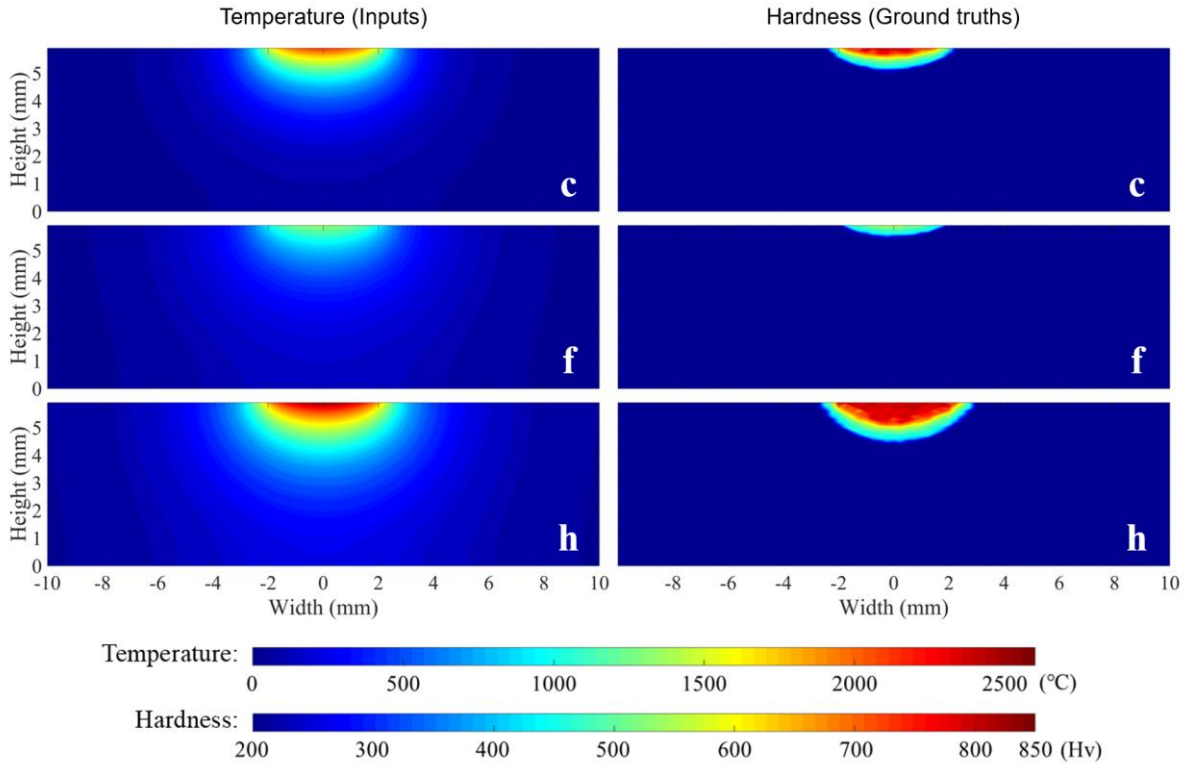

Figure 2.6. Temperature distributions (inputs to the model, left figures) and the corresponding hardness distributions (ground truths, right figures) in the training data set shown side by side (*y*-*z* plane).

Because of the very small size of the training data set, a data augmentation method was necessary to properly train a deep learning model. In this study, a data augmentation method which is frequently used in computer vision [55] was employed. Instead of using one entire image as one data point, a total of 71 smaller images were generated from one image by successively capturing 5.9 mm × 6 mm square regions that were separated by a distance of 0.2 mm (**Figure 2.7**). Therefore, from a training set consisting of only three data, a total of 213 training data points (213 temperature distributions and corresponding 213 hardness distributions; used in training as a pair) were produced.

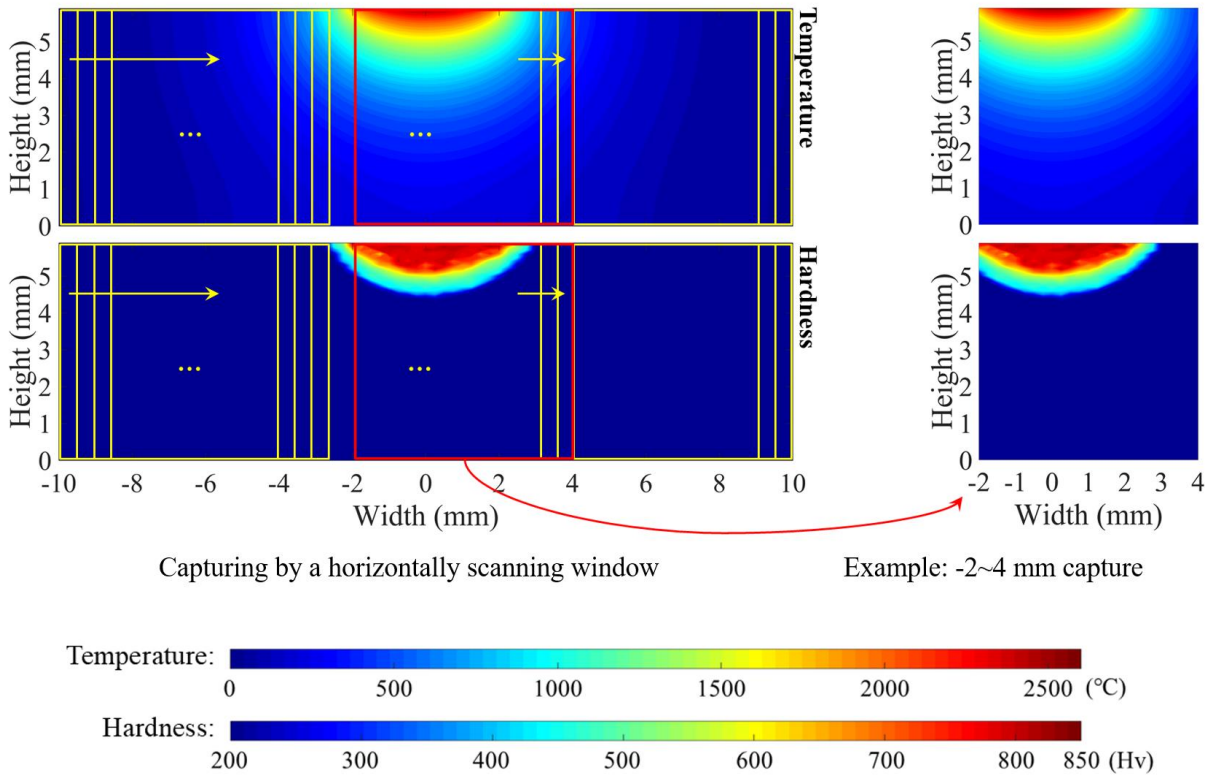

Figure 2.7. A process of generating many smaller image data from one entire image by capturing 5.9 mm × 6 mm regions successively as the red square window moves from left to right, with a separation distance of 0.2 mm. An example pair of temperature and hardness images is shown at the right.

2.3 Deep Learning Model

In this study, cGAN was used as the main framework for temperature-to-hardness translation (**Figure 2.8**) [36, 38, 52]. It is composed of two major components, a generator and a discriminator, which were implemented using deep neural nets based on CNN. All the deep learning architectures in this study were implemented using Python™ and TensorFlow™.

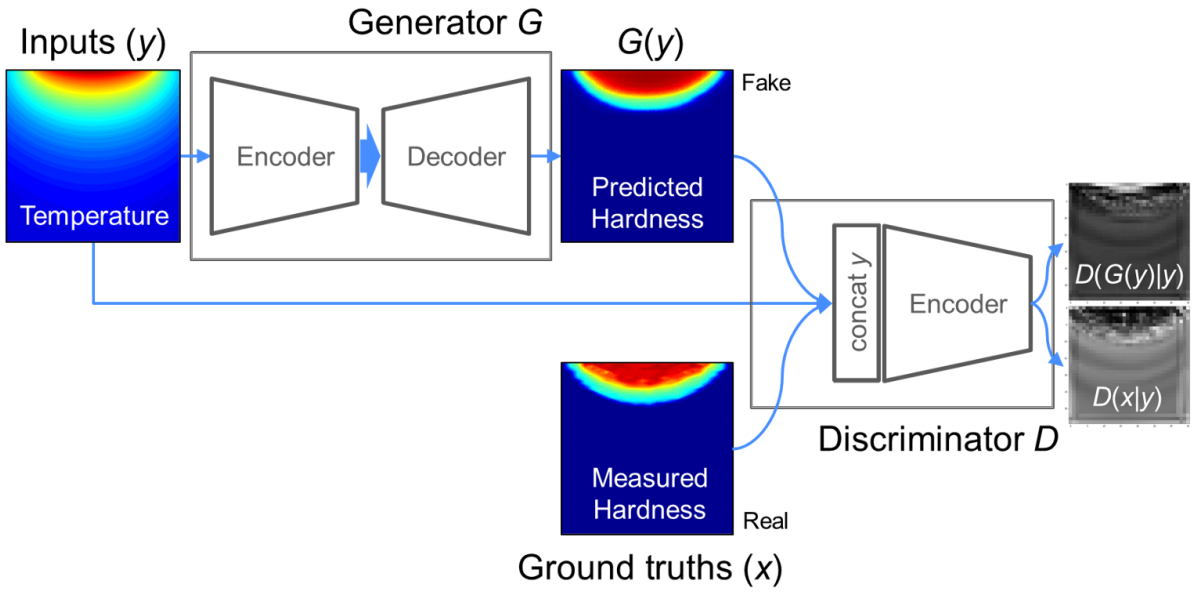

Figure 2.8. Flow diagram for the applied deep learning model.

2.3.1 A temperature-to-hardness generative model

Figure 2.9 presents a flow diagram of cGAN at the training and test stages. In the training stage, the generator and the discriminator are trained using the supplied training data pairs (temperature and hardness), and in the test stage, the trained generator makes predictions by mapping a temperature input to a hardness distribution. In this study, both the generator and the discriminator were designed using CNN, and they recognized temperature and hardness distributions as image tensors consisting of 296 × 301 pixels ([H, W, C] = [296, 301, 1]), where one pixel is 20 μm × 20 μm in size. (H and W denote the number of pixels in the height and width, respectively, and C represents the number of channels.)

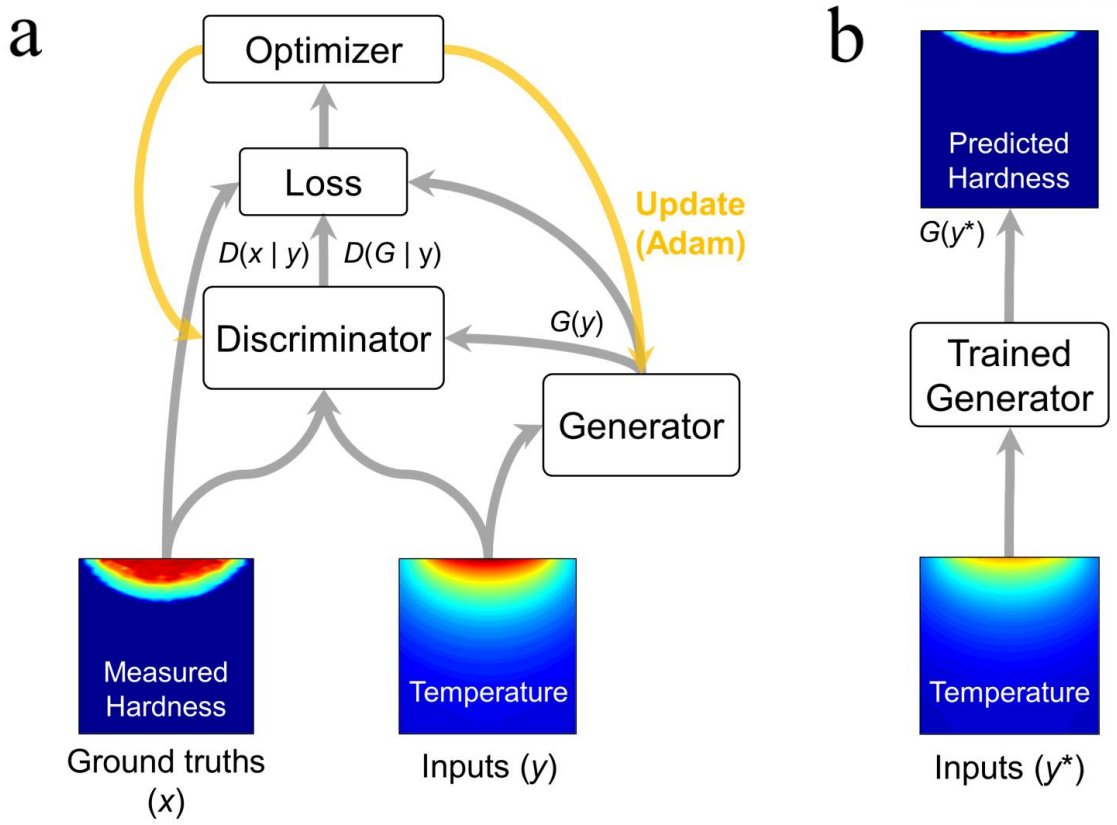

Figure 2.9. Flow diagram of cGAN (a) at the training stage and (b) at the test stage.

In the training stage, as shown in **Figure 2.8** and **Figure 2.9(a)**, cGAN comprises two main components, the generator (G) and the discriminator (D). The generator receives an input temperature image as a condition (y) and learns to translate it to a hardness map $G(y)$ via a deep convolutional network. The translated (or generated) hardness map $G(y)$ is then sent to the discriminator. The discriminator, which is also implemented as a deep convolutional network, receives both of the ground truth image (x, measured hardness) and the generated image ($G(y)$, fake), and determines whether $G(y)$ is real or fake. The generator and the discriminator in cGAN are written mathematically as [36, 52]

$$G = G(z|y;\theta^G),\ D = D(\text{input image}|y;\theta^D), \tag{2.4}$$

where z is a random latent variable, and θ^G and θ^D are trainable parameters in the deep neural nets of the generator and the discriminator, respectively (convolutional filters and bias). In this study, to rule out the possibilities of multiple generations, the random latent variable z was removed and only y was used as the generator ($G = G(y;\theta^G)$) input. The discriminator input was conditioned by the corresponding

temperature distribution, as shown in **Figure 2.8**, so that the discriminator would learn that they were paired. Therefore, the conditioning is simply a concatenation of the images in the channel direction.

The discriminator first receives measured hardness distributions x and is taught that they are real (trained as $D(x|y)\rightarrow 1$). Then, hardness distribution images created by the generator are sent to the discriminator with an indication that they are fake (trained as $D(G|y)\rightarrow 0$). As shown in **Figure 2.11**, the discriminator outputs, which are between 0 and 1, are implemented with a sigmoid activation function (sigmoid(x) = $1/(1+e^{-x})$). The cost function for the discriminator can be expressed mathematically as

$$J^D(\theta^D,\theta^G) = -\frac{1}{2}\mathbb{E}_{x,y\sim p_{data}(x,y)}[\log D(x|y)] - \frac{1}{2}\mathbb{E}_{y\sim p_y}[\log(1-D(G(y)|y))], \quad (2.5)$$

where J^D is a binary cross entropy loss for D and is a function of both θ^D and θ^G; p is a probability density function of the incoming images; and $\mathbb{E}$ is the expectation [52, 56]. For example, $\mathbb{E}_{x,y\sim p_{data}(x,y)}[\log D(x|y)]$ is the expectation of $\log D(x|y)$ with a given x and y, which are distributed by $p_{data}(x,y)$. The discriminator tries to minimize the cost function $J^D(\theta^D,\theta^G)$ (*i.e.*, maximize $D(x|y)$ and minimize $D(G|y)$) while updating only θ^D, the trainable parameter for D.

The cost function for the generator J^G is given as [52, 56]

$$J^G(\theta^D,\theta^G) = -\mathbb{E}_{y\sim p_y}[\log(D(G(y)|y)], \quad (2.6)$$

and the generator attempts to maximize $D(G|y)$ (*i.e.*, $D(G|y)\rightarrow 1$) while updating only θ^G. This implies that the generator is trained to create the most sophisticated and realistic output images that are similar to the actual images to fool the discriminator. In other words, the distribution of the generated images by the model (p_{model}) approaches p_{data}, the distribution of real images. Meanwhile, both cost functions take parameters of the other but cannot update them, as written in **Equation 2.4**. D updates its parameters θ^D to minimize J^D, and G updates its parameters θ^G to minimize J^G, simultaneously.

To sum up, the training process of GAN is a two-player minimax game with a value function $V(D,G)$ [36, 52]:

$$V(D,G)=\mathbb{E}_{x,y\sim p_{data}(x,y)}[\log D(x|y)]+\mathbb{E}_{y\sim p_y}[\log(1-D(G(y)|y))], \tag{2.7}$$

$$\begin{aligned} &D\text{'s objective}: \max_D V(D,G)\ (G \text{ is fixed}),\\ &G\text{'s objective}: \min_G V(D,G)\ (D \text{ is fixed}) \end{aligned}. \tag{2.8}$$

With the cGAN losses, in this study, an L1 loss was added to J^G, which is the absolute difference between the real hardness (x) and the generated hardness ($G(y)$), so the final objective of the generator is written as

$$\tilde{J}^G(\theta^D,\theta^G)=-\mathbb{E}_{y\sim p_y}[\log(D(G(y)|y)]+\lambda\mathbb{E}_{x,y\sim p_{data}(x,y)}[\|x-G(y)\|_1], \tag{2.9}$$

where λ is a coefficient for the L1 loss (set to 100). According to pix2pix [38], the addition of an L1 loss to the cGAN loss function results in improved performance, because the added L1 loss contributes to bridging the difference in pixel values between $G(y)$ and x directly, and thus helps initiate the generator's training, while the original cGAN losses focus on catching the high-level information contained in real images, such as naturalness, texture and style.

2.3.2 Network architecture

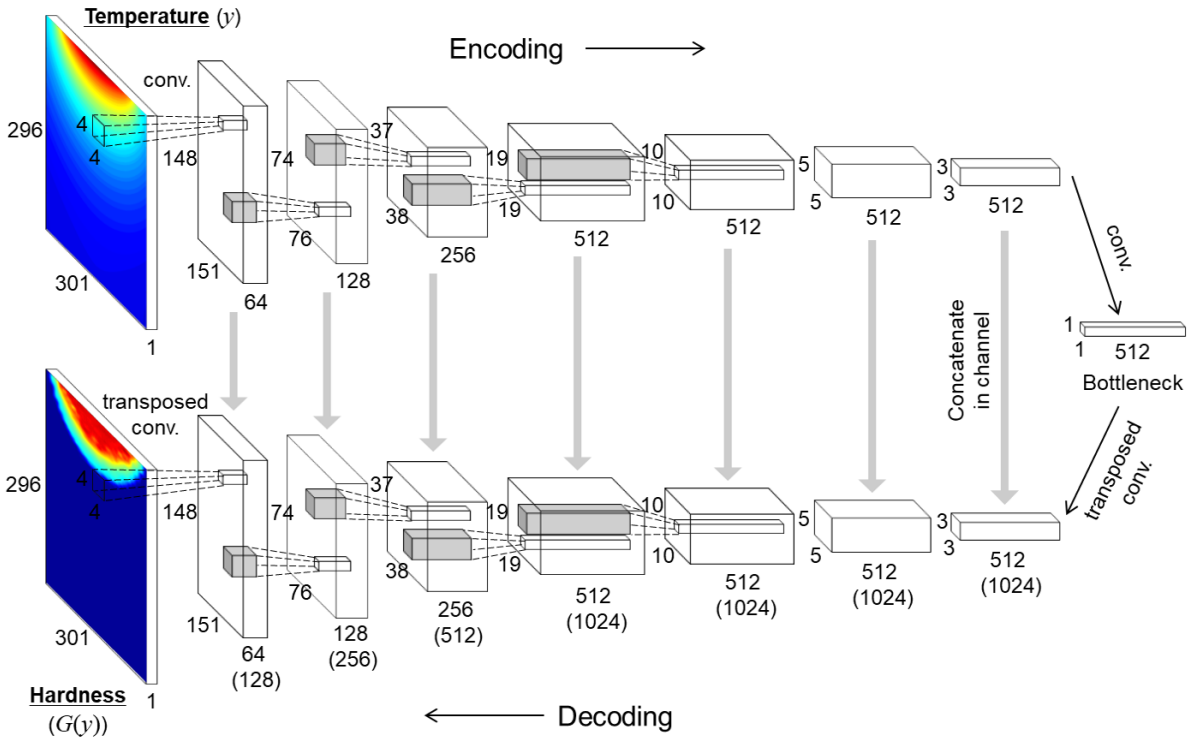

Figure 2.10. Generator structure: A CNN-based encoder-decoder with skip connections from the U-Net. The numbers in parentheses are the channel numbers after concatenations.

Figure 2.10 presents the detailed network structure of the generator, where a convolutional process between two successive layers is depicted schematically using dashed lines connecting two rectangular parallelepipeds. The three numbers around each layer represent H, W, and C. In **Figure 2.11**, discriminator's structure is drawn in the same way. In **Table 2.3**, detailed information on the network architecture is summarized.

In the generator, the input temperature distribution was encoded via eight convolutional layers in the encoding part (the upper row in **Figure 2.10**, from left to right), and in the decoding part (the lower row in **Figure 2.10**, from right to left), a low resolution feature map (a pixel) at the bottleneck of the shape [1, 1, 512] was reconstructed to an input level resolution hardness map through eight transposed-convolutions [57], which are reverse processes with respect to the convolutions. Note that, as reported [58, 59], highly-advanced information contained in the input image can be extracted as the convolutional network becomes deeper. The filter height and width were both four (FH = FW = 4) and stride = 2 except at the bottleneck (stride = 3) and the last two layers in the discriminator (stride = 1) (See **Table 2.3**).

Skip connections of the U-Net [60] were utilized in the generator as visualized by gray arrows in **Figure 2.10**. The encoded temperature images were concatenated to the corresponding decoders, and by receiving encoded information from the encoders directly, the decoder became less dependent on the bottleneck layer, and more stable up-sampling was possible. In addition, a better-quality image was generated with the U-Net because each encoded feature map could be instantly used for the next level up-sampling. For the discriminator, a CNN-based encoder was employed to extract the features contained in the hardness and temperature maps (**Figure 2.11**) and examine the encoded feature map on patches [61].

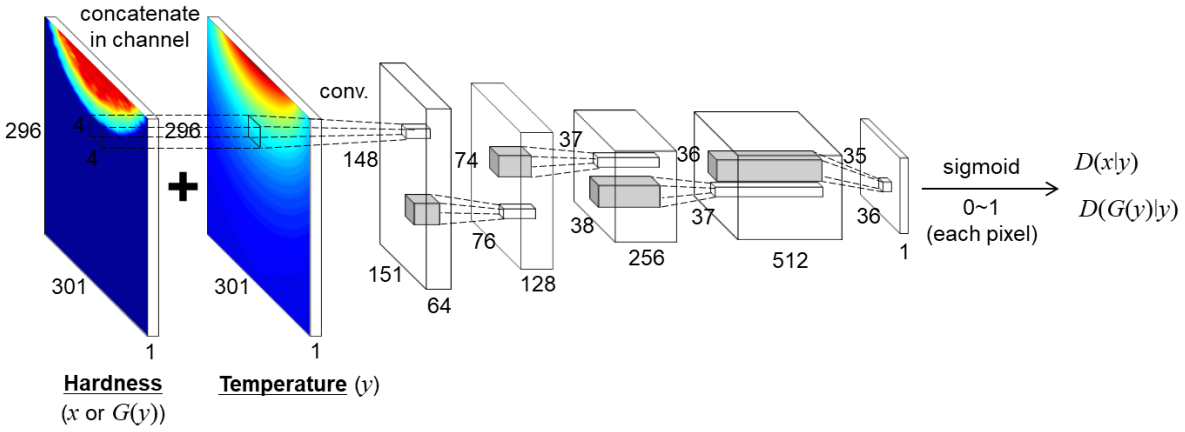

Figure 2.11. Discriminator structure: A CNN-based encoder, examining the inputs on patches.

In this study, after the convolutional layers, a batch normalization (BN) layer [62] and an activation layer were added, following the guidelines proposed in deep convolutional GAN (DCGAN) [37]. A batch normalization layer has the effect of improving learning speed, suppressing overfitting, and reducing dependency on the initial values of learning variables, such as the weights in the filters. In this layer, a normalization process is performed so that the input data have a mean of zero and a variance of one for each channel in a minibatch, and the values become appropriately distributed, so that major problems in machine learning (such as gradient vanishing and gradient exploding) can be prevented. The process was used in every layer, except for the first and the last layers in both the generator and discriminator. After the batch normalization, to provide nonlinearity and for the model to represent more rich features, a rectified linear unit (ReLU [63]; $\max(0, x)$) function and a leaky ReLU (LReLU [64]; $\max(0.2x, x)$) were added as activation layers in the decoders and encoders, respectively, except for the output layers in both the generator and discriminator, which used hyperbolic tangent and sigmoid functions, respectively.

Table 2.3. Detailed network architectures for (a) the generator and (b) the discriminator. LReLU, conv., BN, concat, and t-conv. denote leaky ReLU, convolution, batch normalization, concatenation, and transposed-convolution, respectively.

(a) Generator	(b) Discriminator
Temperature distribution $y \in \mathbb{R}^{H \times W \times 1}$, [-1, 1] (normalized)	Generated hardness $G(y) \in \mathbb{R}^{H \times W \times 1}$, [-1, 1] or Measured hardness $x \in \mathbb{R}^{H \times W \times 1}$, [-1, 1] (normalized)
stride=2 conv. 64 (h_0)	concat y, stride=2 conv. 64
LReLU, stride=2 conv. 128, BN (h_1)	LReLU, stride=2 conv. 128, BN
LReLU, stride=2 conv. 256, BN (h_2)	LReLU, stride=2 conv. 256, BN
LReLU, stride=2 conv. 512, BN (h_3)	LReLU, stride=1 conv. 512, BN
LReLU, stride=2 conv. 512, BN (h_4)	LReLU, stride=1 conv. 1
LReLU, stride=2 conv. 512, BN (h_5)	sigmoid
LReLU, stride=2 conv. 512, BN (h_6)	
LReLU, stride=3 conv. 512, BN (h_7)	
LReLU, stride=3 t-conv. 512, BN (h_8)	
concat h_6, ReLU, stride=2 t-conv. 512, BN (h_9)	
concat h_5, ReLU, stride=2 t-conv. 512, BN (h_{10})	
concat h_4, ReLU, stride=2 t-conv. 512, BN (h_{11})	
concat h_3, ReLU, stride=2 t-conv. 256, BN (h_{12})	
concat h_2, ReLU, stride=2 t-conv. 128, BN (h_{13})	
concat h_1, ReLU, stride=2 t-conv. 64, BN (h_{14})	
concat h_0, ReLU, stride=2 t-conv. 1 (h_{15})	
tanh ($G(y)$)	

2.3.3 Training details

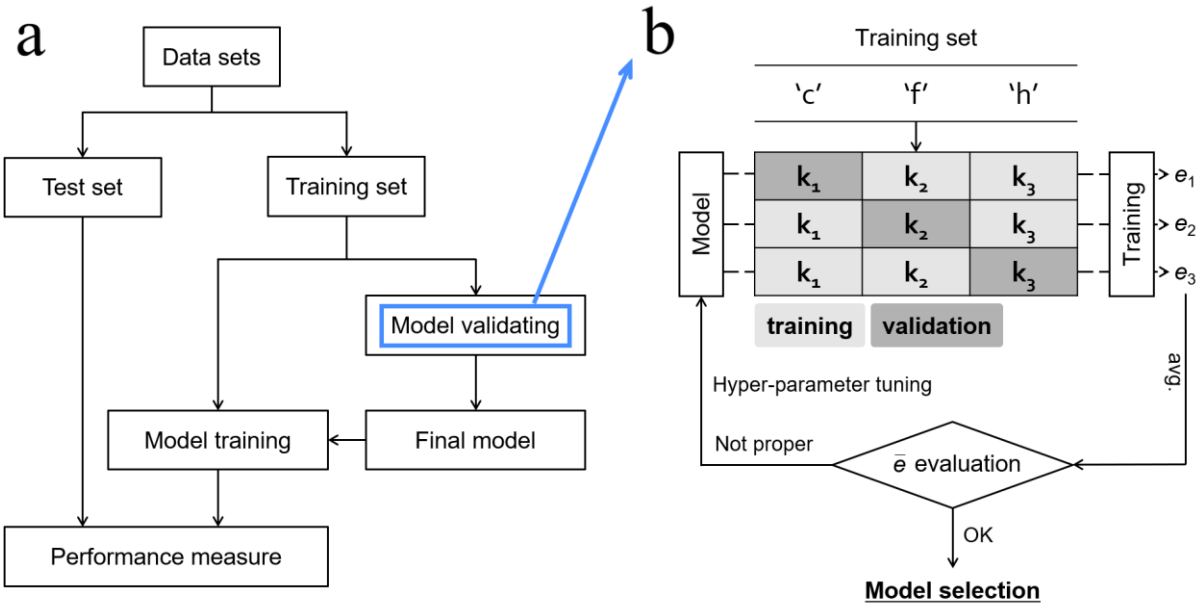

Figure 2.12. (a) A flow chart showing the training, validation, and performance measure processes. (b) A 3-fold cross-validation process.

Figure 2.12 shows the overall training and validation processes. Hyper-parameters such as batch size, learning rate, and total epochs were determined under a *k*-fold cross-validation. The whole data set was divided into the training set and the test set, and the training set was used for the cross-validation, as shown in **Figure 2.12**. During the cross-validation, among the pre-described three training process conditions (c, f, h), two conditions were used in the training and the other condition was applied for the validation; therefore, a total of three sessions were made, as shown in **Figure 2.12(b)**. Then the training and the validation errors e_1, e_2, and e_3 from the three training processes were averaged, and used for the tuning of hyper-parameters. With the tuned hyper-parameters, the final model was trained using all the training set data, and the test set data were used only to measure the performance of the final model and not for any training or validation processes. In **Figure 2.13**, a comprehensive flow chart for the training is given.

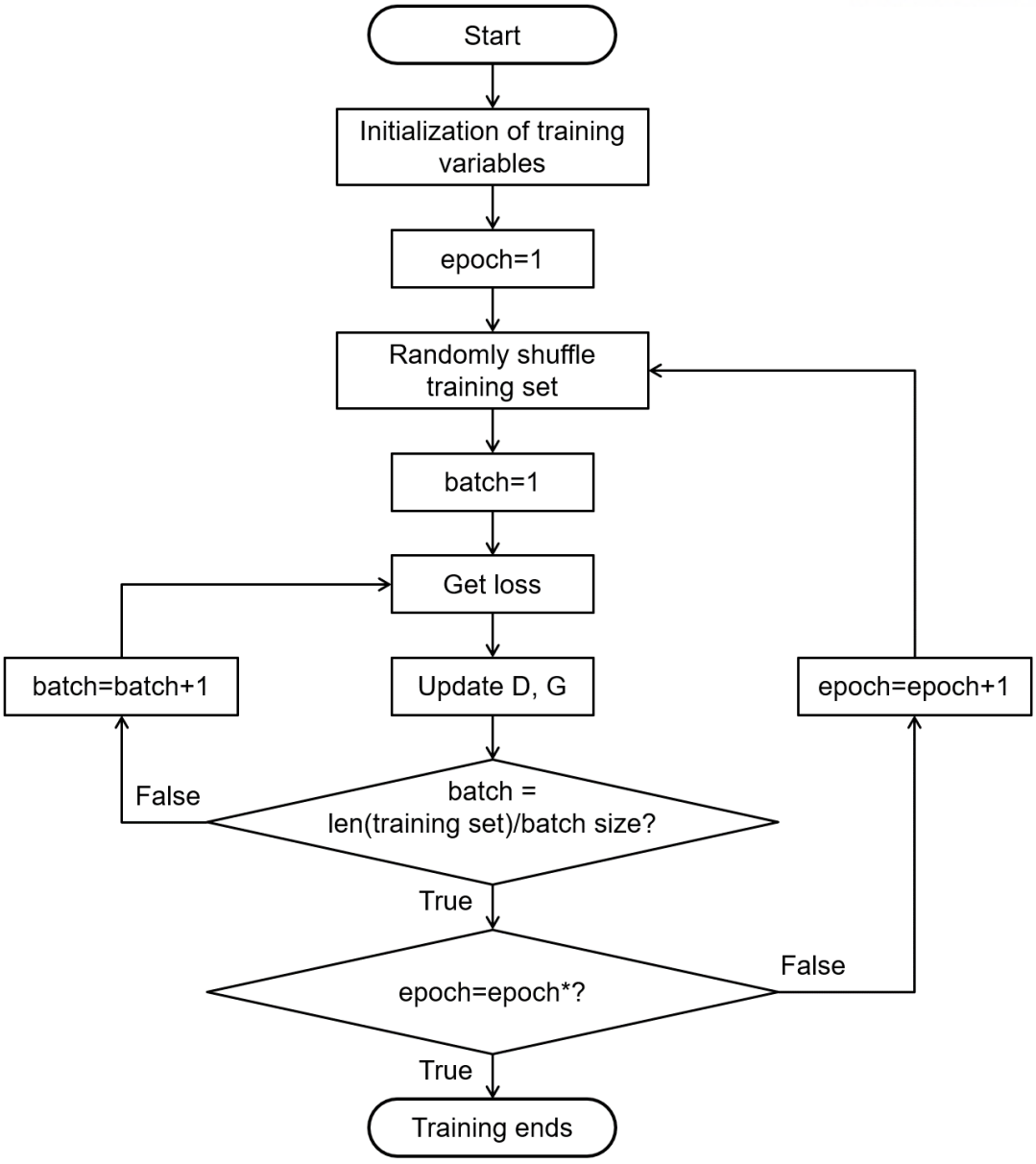

Figure 2.13. A comprehensive flow chart describing the training. len(training set) is the number of temperature-hardness pairs in the training set.

To train the presented deep learning model with the given cost functions, the adaptive moment estimation (Adam) [65] optimizer was employed to update the training parameters by backpropagation, with β_1 =0.5 and β_2 = 0.999, which are exponential decay rates for the first and second moment estimates, respectively. A batch size of 2 and a learning rate of 0.00001 were used, which were tuned through a grid search. The training data were normalized to be in the range (-1, 1) before training, and randomly shuffled in every epoch. The filters were initially distributed following a normal distribution with a mean of 0.0 and a standard deviation of 0.02 [37], and the bias was initialized to zero.

Figure 2.14 presents the root mean square errors (RMSE) of the training and validation processes for the final model plotted against the number of epochs. As shown, the minimum validation error was achieved at epoch 461, so the training was stopped there to avoid overfitting (thus, for the training the final model, epoch* = 461 in **Figure 2.13**). The corresponding L2-norm error for the training was 0.549% (with an R^2 value of 99.95%).

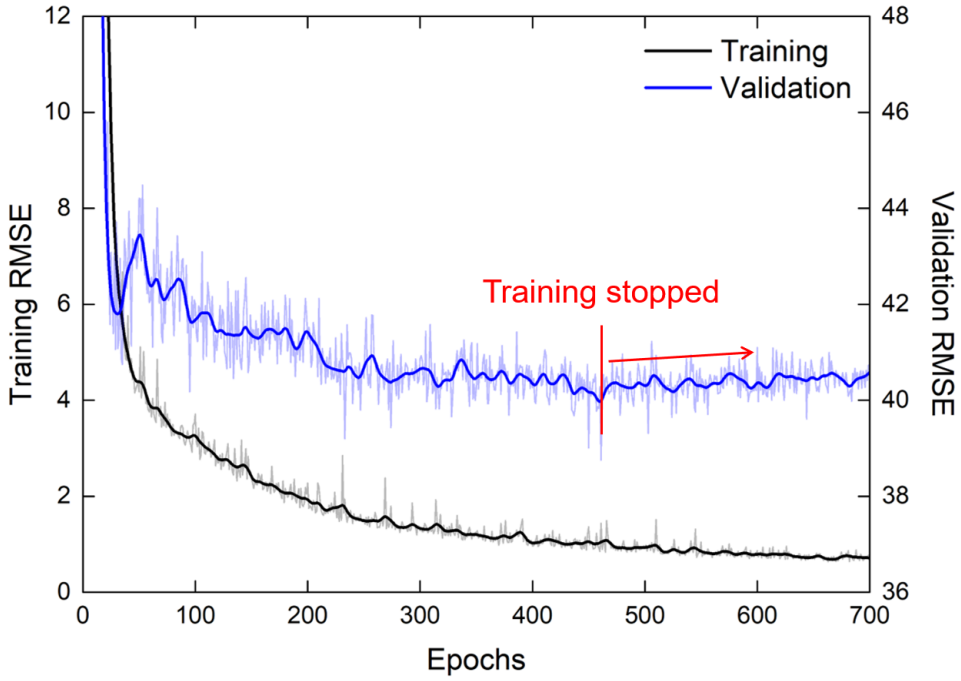

Figure 2.14. RMSE histories of the final model according to the number of epochs. The training and validation errors are represented by black and blue lines, respectively. The thick lines are trend lines. Training was stopped at epoch 461.

2.4 Results and Discussion

After validation, the deep learning model was trained using the entire training data. An example showing how the images in the training data set were processed through the convolutional layers of the generator and discriminator is given in **Figure 2.15** and **Figure 2.16**, respectively (data set h at epoch 20). Only the first channel (C = 1) was shown. Relatively early stage results were chosen to clearly show the differences between the two kinds of discriminator inputs (real and fake) before the model was fully trained and to observe how images were processed in the generator in an early stage of the training. In the generator, the feature maps in each layer were denoted by h (h_0~h_{15}; see **Table 2.3**).

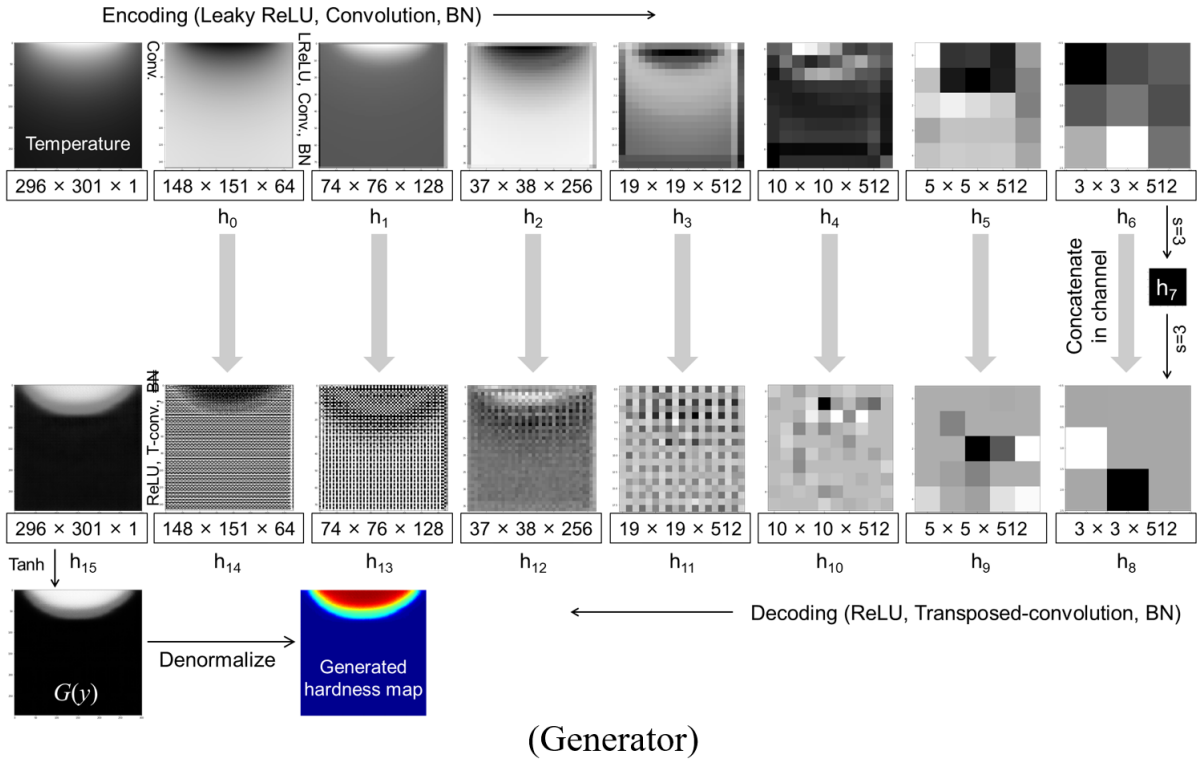

Figure 2.15. Insides of the generator (epoch = 20). This result is for data set h, and only the first channel (C = 1) is shown for brevity.

In **Figure 2.15**, the top row (encoding line) shows how the features optimized for translating a temperature distribution to a corresponding hardness map are extracted through a series of convolutional layers. For example, in the h_1 layer, a region looking like a HAZ is distinguishable from other weakly heated regions, and in the h_2 layer, after receiving feature maps from the h_1 layer, the HAZ seems to be divided into more hardened and less hardened regions. Although the feature maps deeper

inside the network have low resolutions and the operations there seem abstract, it is believed that optimal information is successively extracted layer after layer, such that the temperature distribution can be transformed to a hardness map properly. The bottom row (decoding line) in **Figure 2.15** shows how the extracted features are reconstructed to an input-level, high-resolution hardness map.

In **Figure 2.16**, the encoding process in the discriminator is presented. The discriminator encodes the features optimized for judging the authenticity of the inputs on patches. Feature maps generated from the measured (real) hardness are presented in the top row, and those from the generator (fake) in the bottom row. Because the presented features are from an early stage of the training (*i.e.*, the generator is not yet fully developed), $D(x|y)$ and $D(G|y)$ show a large difference. After a sufficient number of epochs, however, the generator was able to create realistic images, and the discriminator barely distinguished the real and fake inputs.

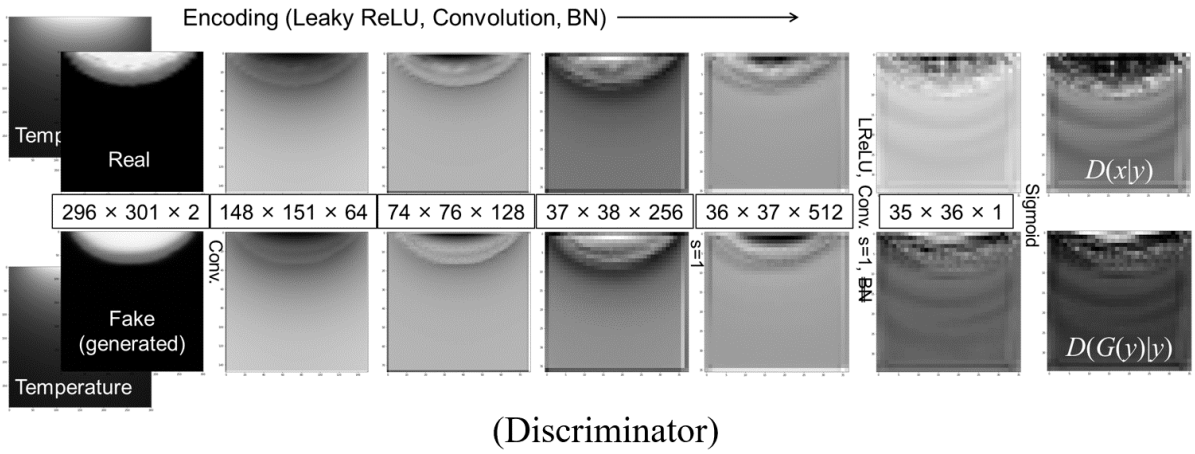

Figure 2.16. Insides of the discriminator (epoch = 20). This result is for data set h, and only the first channel (C = 1) is shown for brevity.

After validation and final training of the model are complete, the test dataset is input (a, b, d, e, and g), and the predicting processes for all five conditions are presented in **Figure 2.17**. For each process condition, both encoding (left to right) and decoding (right to left) processes are shown. (Bottleneck layer is omitted and only the first channels are shown in the figure.)

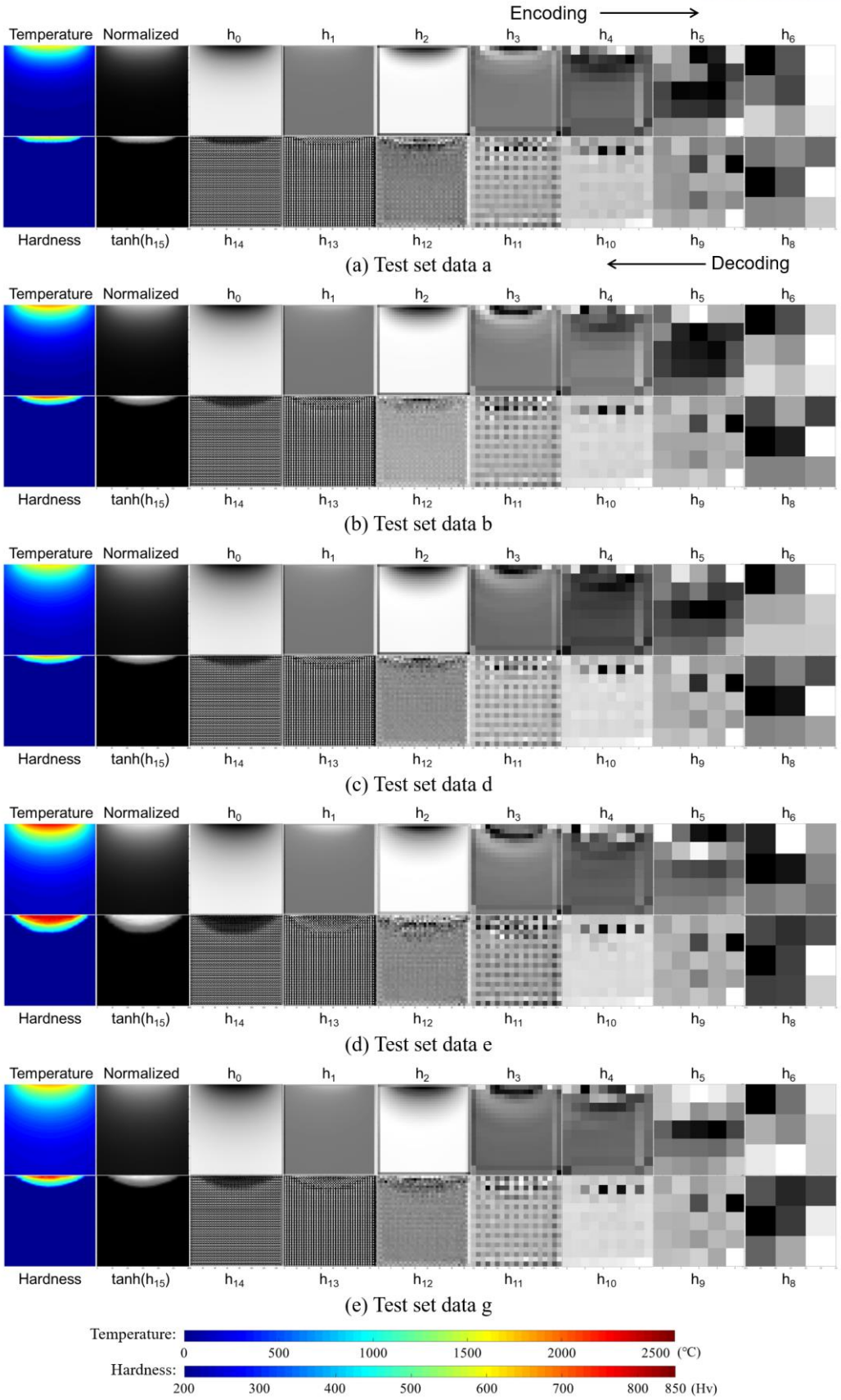

Figure 2.17. Insides of the generator in the test mode. From top to bottom, the process parameters are a, b, d, e, and g, respectively. Only the first channel (C = 1) is shown.

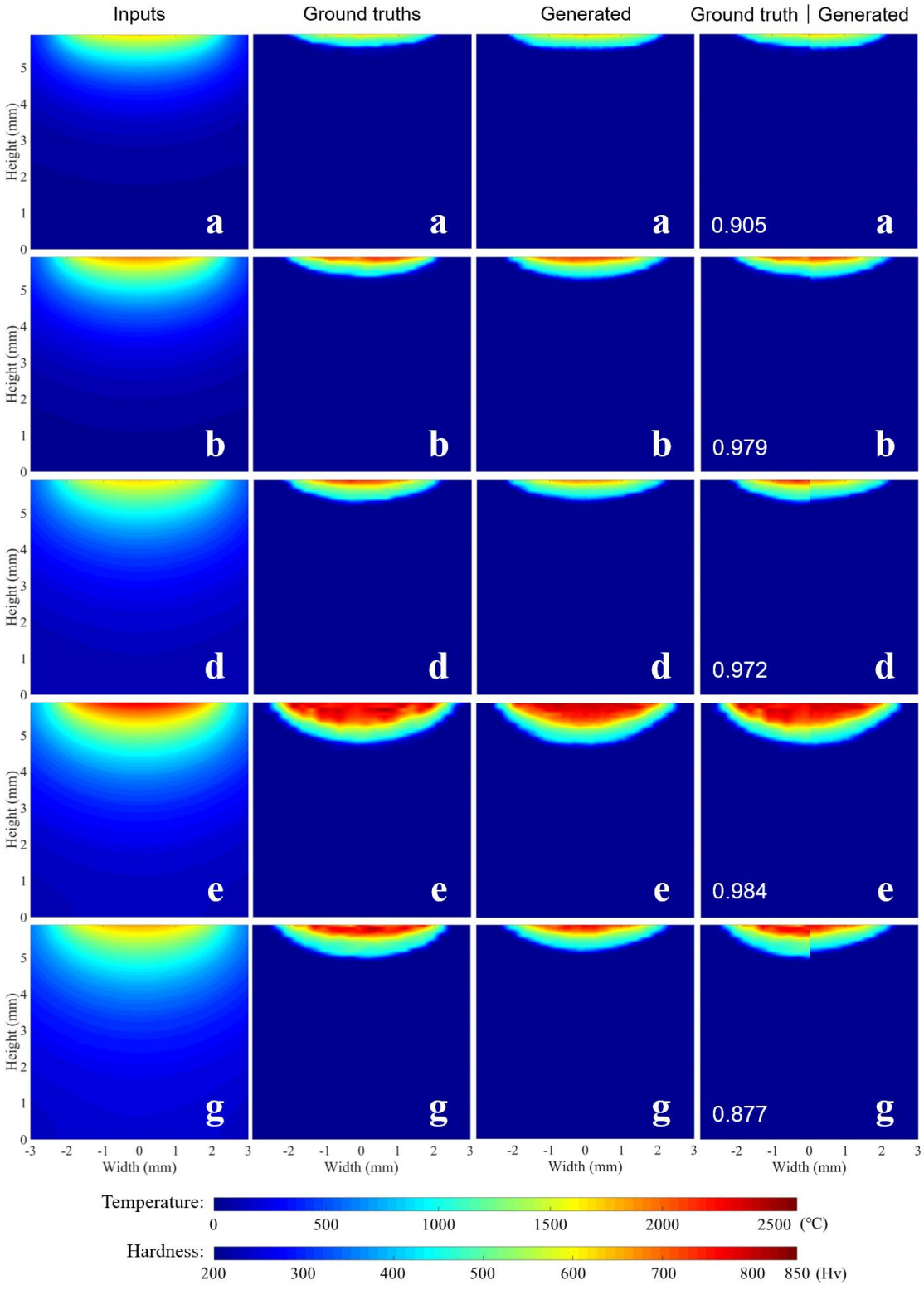

Figure 2.18. Prediction of hardness distribution using the test data set (from top to bottom, a, b, d, e, and g): inputs (temperature distributions), ground truths (measured hardness distributions), generated (predicted) hardness distributions, and measured (left half) and predicted hardness (right half) distributions are shown together in the first, second, third, and fourth columns, respectively, with the R^2 accuracies shown at lower left corners.

Figure 2.18 presents the hardness prediction results using the five test data sets (a, b, d, e, and g) obtained by the developed deep learning model. In this case, hardness maps were created from the test data set without any iterations. The results for a, b, d, e, and g are presented from top to bottom, and from left to right, temperature distributions (inputs), measured hardness distributions (ground truths) and predicted (generated) hardness distributions are presented, and measured (left half) and predicted hardness (right half) distributions are compared together in the last column, with R^2 accuracy values shown in the lower left corners. As shown, the generator successfully predicted hardness distributions that match very well with the measurement data, and the R^2 accuracy varied between 0.877 and 0.984.

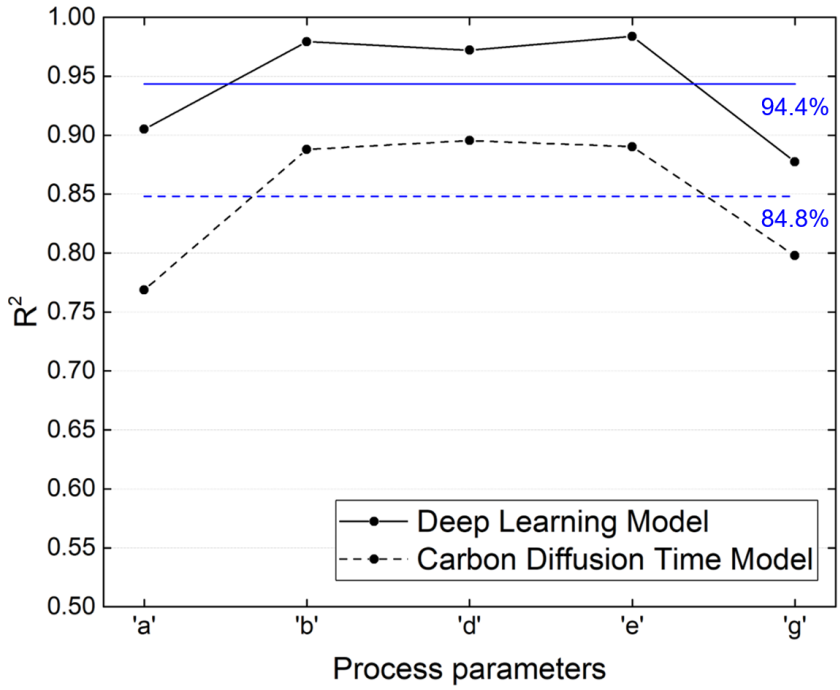

Figure 2.19. Accuracy (R^2) comparison with the carbon diffusion time model for all five conditions of the test data set. Average accuracy values for both models are shown using blue horizontal lines (solid and dashed).

To compare the prediction accuracy of the deep learning model, the R^2 values of the prediction results from the carbon diffusion time model [1] were calculated and are shown together in **Figure 2.19**. (average accuracy values for both models are shown using blue horizontal lines.) The accuracy of the deep learning model (average: 94.4%) is substantially better than that of the carbon diffusion model (average: 84.8%) for all five process conditions. An accuracy value close to 95% is believed to be very good (considering only three process conditions were used for the training), which in the authors' opinion shows the capabilities of deep learning very well.

2.5 Conclusion

A data-driven deep learning model was introduced to predict a hardness distribution in the laser heat treatment of AISI H13 tool steel from a temperature distribution obtained by a 3-D thermal simulation. Experimentally-measured hardness distributions on the cross-section of heat-treated specimens were provided to the network as ground truths. The major findings of this study can be summarized as follows:

- A data set consisting of only three process conditions was used for the training. By employing a scanning window, the data set was effectively augmented to 213 data points, with which reasonably good predictions were possible.

- A cGAN architecture with a CNN-based encoder-decoder was employed for the model, and the model's cost function was combined with an L1 loss. It was shown that, with the L1+cGAN cost, the quality of the image-to-image translation (from temperature to hardness) was sufficiently good.

- As in the face recognition deep neural networks, which can identify facial expressions or identities through the shapes of eyes or mouths with high accuracies, the proposed deep learning model successfully recognized the features optimized for converting temperature to hardness, such as highly-heated areas. The average accuracy of the predictions was 94.4%, which the authors believe is fairly good.

Chapter 3. Deep Learning Study on Laser Keyhole Welding

This chapter includes the published contents:

3.1 Introduction

Laser welding demonstrates high strength, narrow and deep bead shapes, but input laser processing parameters (such as laser power and beam scanning speed) must be carefully selected to achieve the desired weld bead, as the weld bead properties (shape, heat affected zone, microstructures, porosity, etc.) substantially affect the key mechanical properties such as tensile strength, ductility, hardness and fatigue [66]. Therefore, a reliable predictive model is essential, especially for challenging welding applications.

Many researchers have endeavored to establish a good predictive model, as laser welding has become increasingly in demand for a variety of cutting-edge technologies. Lankalapalli *et al.* [67] developed a penetration depth predictive model in terms of laser power and Péclet number, by solving a two-dimensional heat conduction equation with a conical keyhole shape assumption. Lampa *et al.* [68] predicted the penetration depth and weld width using a simplified thermal model, by introducing an effective thermal conductivity considering thermocapillary flow. Chang and Na [69] proposed a new volumetric heat source equation and analyzed the effect of heat source descriptions in laser micro welding. Ki *et al.* [19, 20] developed a self-consistent laser keyhole welding model by considering various physical phenomena at the liquid-vapor interface. Benyounis *et al.* [70] conducted a response surface methodology analysis to achieve correlations for the weld bead geometry in terms of laser power, scanning speed, and focal point position. Hann *et al.* [71] reported a simple physical model to predict the melt depth and width using mean surface enthalpy values. Volpp and Vollertsen [72] established an analytical model of multiple reflections, based on a ray tracing method. Using the two types of beam profiles, keyhole shapes were predicted and compared to the experimental results. Courtois *et al.* [21] developed a laser keyhole welding model using COMSOL Multiphysics® software, and they considered the interaction of laser beam and melt pool using the Maxwell equations. Kim and Ki [73] presented a simple scaling law for predicting penetration depth according to laser processing parameters by considering the strength of multiple reflections. Suder and Williams [74] reported an empirical model for predicting penetration depth in terms of laser power, interaction time, and beam diameter.

Fabbro [75] studied a scaling law regarding the keyhole depth based on thermal dimensionless parameters. While most of the referenced studies have been useful, they normally require high computational cost with a long computation time and the focus was primarily limited to the geometrical shape of the bead (mostly penetration depth).

In this study, for the first time, a novel deep learning framework was proposed for predicting OM image of the cross-sectional laser weld bead, from only two laser processing parameters (laser intensity I_0 and beam interaction time t_i). Our deep learning model can predict the weld bead in real image, *i.e.* including keyhole, heat affected zone, substrate, microstructures, porosity as well as the geometrical bead shape, which synthetically determine the mechanical properties and weld quality. Also, it can instantly generate multiple predictive bead images in a few seconds from the given input laser process conditions once training ends, so is very handy as well as practical (one can also share the trained model online, using the open source deep learning libraries such as TensorFlow and PyTorch on GitHub). Note that several approaches using an ANN have been reported in bead shape prediction in laser welding [76-78], however, to the best of the authors' knowledge, no deep learning model has been reported yet.

The proposed deep learning model is composed of two successive generators, because prediction of such a high-resolution OM image using only one generator is a tough job (the information contained in the input laser process conditions is very limited to generate the OM image). In the first generator, an encoder–decoder network [79] based on the CNN [35] was adopted that converts the input laser processing parameters into a weld bead segmentation map. In the second generator, for a guided high-resolution OM image synthesis, pix2pixHD [40] was adopted. It was basically a CNN-based encoder–decoder on a cGAN [36, 52], with multi-scale generators and discriminators, and deep residual network [80, 81] which is the structure widely used in a super-resolution (SR) problem [41, 82-85]; thus, the input segmentation map was filled and exhibited a high-resolution OM image resembling a real image. A schematic diagram for the developed AI architecture is given in **Figure 3.1**.

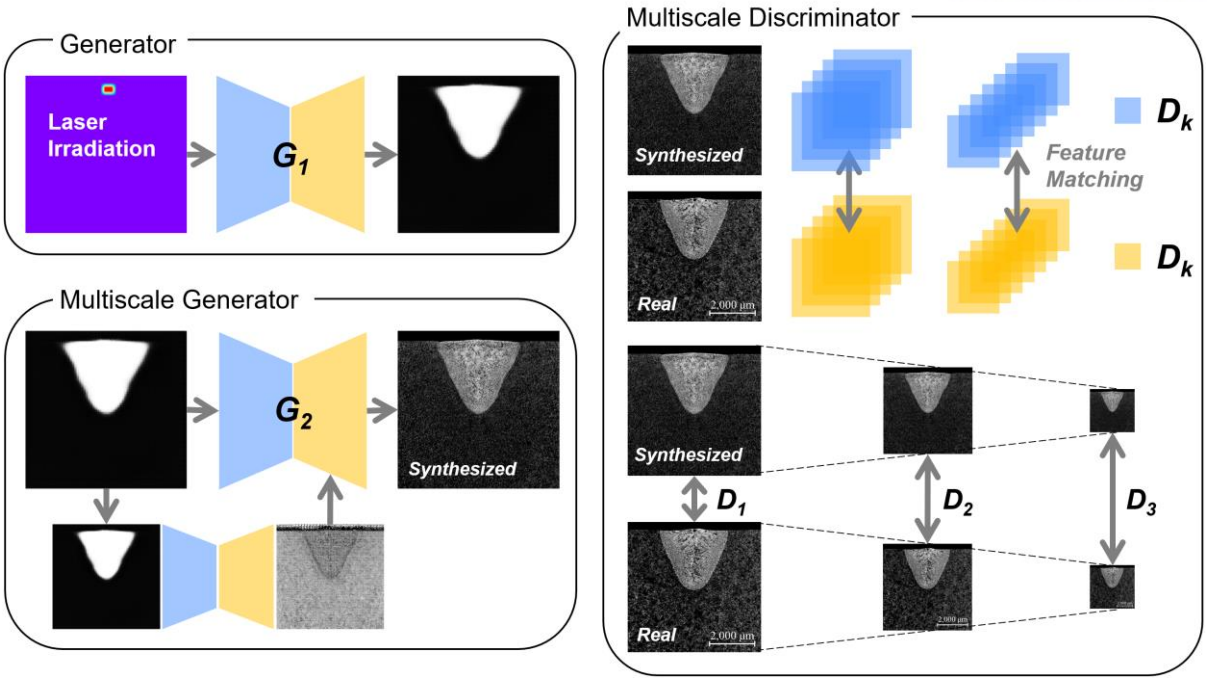

Figure 3.1. Schematic for the AI model developed in this study. It consisted of encoder-decoder generator for the bead shape prediction, and multiscale generator and discriminator for the high-resolution synthesis of optical microscopic image.

In this study, at first, the datasets were obtained from laser welding experiments with AISI 1020 carbon steel using a 2 kW multi-mode fiber laser on a large process window. Then, the datasets were augmented using several image processing techniques, and used for the model training. After model validation, performance of the model was tested using a test dataset. Specifically, accuracies for the penetration depth (89.0%) and weld bead area (93.6%) were calculated, and quality of the predicted OM images was assessed. In summary, the major contributions in this study are as follows.

- A novel deep learning framework was presented for predicting high-resolution cross-section weld bead images in laser welding, from two laser processing parameters (laser intensity and beam interaction time).

- Two individual generators were developed, instead of direct mapping from the input laser process conditions to the high-resolution optical microscopic image, for a stable and semantic learning. A proper data setup method was also proposed accordingly.

- An optimal model structure suitable for the weld bead image prediction was proposed, obtained from the processes of validation and trial and errors. Also, inside of the generators was presented to get a better understanding about the predicting process.

3.2 Data Preparation

In **Figure 3.2**, an overview of the proposed deep learning architecture is presented with data flow pipelines. As shown, three types of data exist (two-channel input of laser processing parameters, weld bead segmentation, and the OM image) and two different generators (G_1 and G_2). In the first generator G_1, which was the CNN-based encoder and decoder (denoted by E and D in the figure, respectively), the weld bead segmentation map was predicted from the two-channel input of the laser processing parameters (I_0 and t_i). Subsequently, in the second generator G_2, which was the CNN-based cGAN structure, the OM image was synthesized from the weld bead segmentation map.

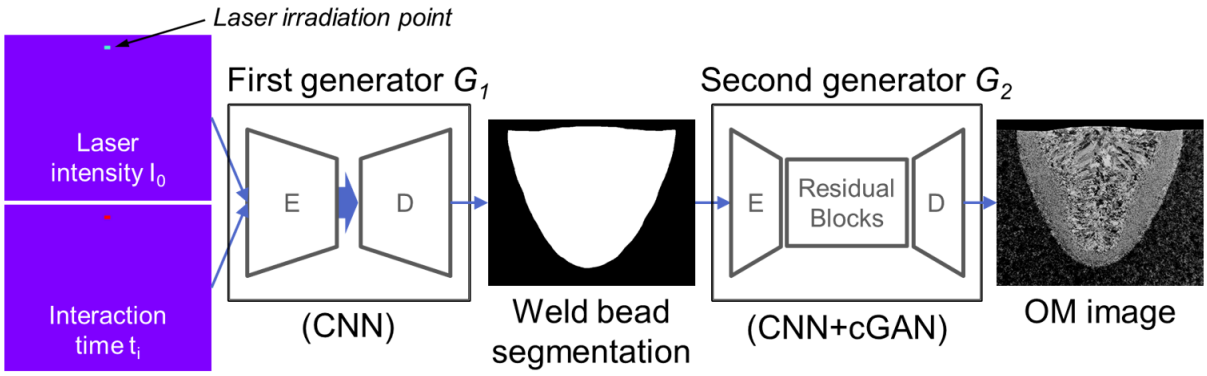

Figure 3.2. Overview of the deep learning model proposed in this study. It consisted of two generators. For the first generator G_1, the weld bead segmentation map was predicted from the two laser processing parameters I_0 and t_i. Subsequently, in the second generator G_2, an OM image was constructed from the weld bead segmentation map. E and D represent the CNN-based encoder and decoder, respectively. In the second generator, the cGAN architecture was additionally employed. The OM image shown is that of 1800 W and 5.313 mm/s.

For the data setup, 45 laser welding experiments were conducted on a large process window, with 45 laser process conditions (I_0, t_i). Note that $I_0 = P/A_{beam}$ and $t_i = D_{beam}/v$, where P, A_{beam}, D_{beam}, and v are the laser power, laser beam area, beam diameter, and laser scanning speed, respectively. After the experiments, a cross-sectional weld bead was observed using optical microscopy; hence, 45 pairs of [(I_0, t_i), OM image] were obtained. Details regarding the experiment are presented in **chapter 3.2.1**.

Next, the laser processing parameters were converted into a two-channel image tensor, as shown in the leftmost of **Figure 3.2**. Moreover, from the OM images, the weld bead maps were segmented manually, such that the pixel value inside the bead was 1.0, and 0.0 elsewhere, as shown in **Figure 3.2**.

Therefore, 45 pairs of [(I_0, t_i), SEG, OM image] are now ready (SEG denotes the weld bead segmentation map). Details are described in **chapter 3.2.2**. Subsequently, the entire data was divided into training, validation, and test sets. To increase the number of training data, the training set was augmented by adopting several image-processing techniques that are primarily used in computer vision [55] (**chapter 3.2.2**). The detailed architectures of G_1 and G_2 are presented in **chapter 3.3**.

3.2.1 Laser welding experiment

To obtain the OM images of the cross-sectional laser weld bead according to the laser process condition (I_0, t_i), a laser welding experiment was conducted, using a 2 kW multi-mode fiber laser (IPG YLS-2000; wavelength of 1070 nm). The laser beam exhibited a circular top-hat intensity profile, and the beam passing through a 200 μm process fiber, 160 mm collimation lens, and 160 mm focusing lens was focused on the surface of the specimen, with a focused diameter of 200 μm. In **Figure 3.3**, a schematic drawing of the experiment is shown with coordinates and dimensions. The specimen was a 30 mm × 40 mm × 12 mm AISI 1020 carbon steel block and its chemical composition is presented in **Table 3.1**.

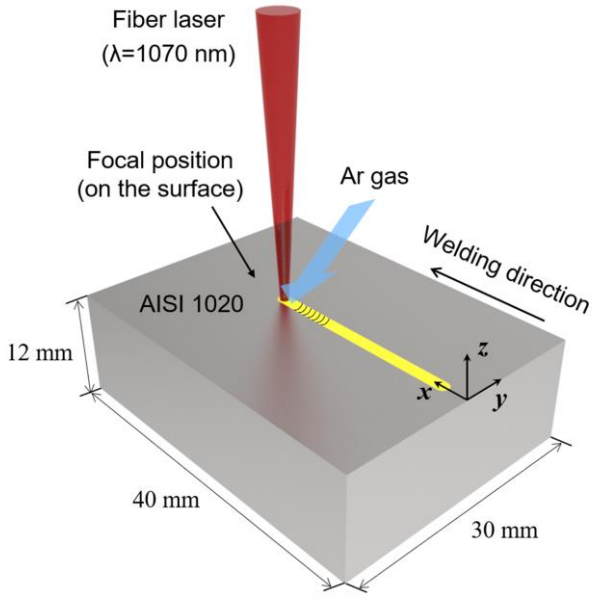

Figure 3.3. Schematic for the laser keyhole welding process.

Table 3.1. Chemical composition of AISI 1020 carbon steel.

AISI 1020	C	Si	Mn	P	S	Fe
(%)	0.185	0.242	0.558	0.0154	0.0051	balance

The specimen surface to be welded (*x*-*y* plane in **Figure 3.3**) was grinded to a roughness of 1 μm and washed with acetone before welding, to remove impurities and cutting fluid remaining on the surface that could affect the beam penetration and surface reflection. During the experiment, the laser head was fixed and the specimen moved in the −*x* direction on a motorized linear stage. Furthermore, as shown in **Figure 3.3**, welding was performed on a bead-on-plate mode, and 25 L/min of argon gas was supplied for shielding. The shielding nozzle was a copper tube with an outer diameter of 10 mm and an inner diameter of 6 mm, and was located with a distance of 1 mm from the specimen surface (*z* direction in **Figure 3.3**) and 1 mm from the laser beam position (*x* direction in **Figure 3.3**). The flow angle was 45°.

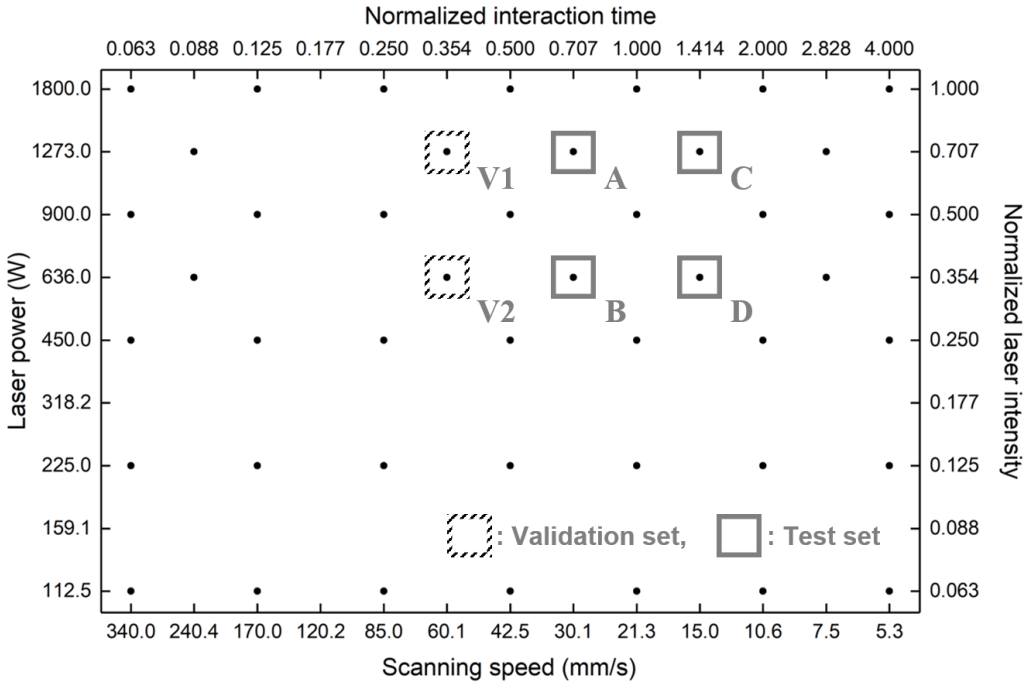

Figure 3.4. Process conditions on a laser power-scanning speed (left *y* – bottom *x*) and normalized laser intensity-normalized interaction time (right *y* – top *x*) planes.

Figure 3.4 shows a process window showing the laser welding conditions on a laser power-scanning speed plane, and as seen, a total of 45 experiments were conducted. Laser power and beam scanning speed were ranged from 112.5 to 1800 W and from 5.313 to 340 mm/s, respectively, on a logarithmic scale. The two process conditions in a dashed square box (marked by V1 and V2) were used for the validation of the deep learning model, and those in a solid square box (marked by A, B, C, and D) were used for the test of the final model (four conditions). The process conditions without any box were applied for the training of the model (39 conditions).

3.2.2 Data preprocessing

In **Figure 3.5**, an example of a two-channel input is presented (process condition: 1800 W and 5.313 mm/s). As shown in the figure, each channel comprises 138 × 142 pixels and the laser irradiation points are magnified on the right.

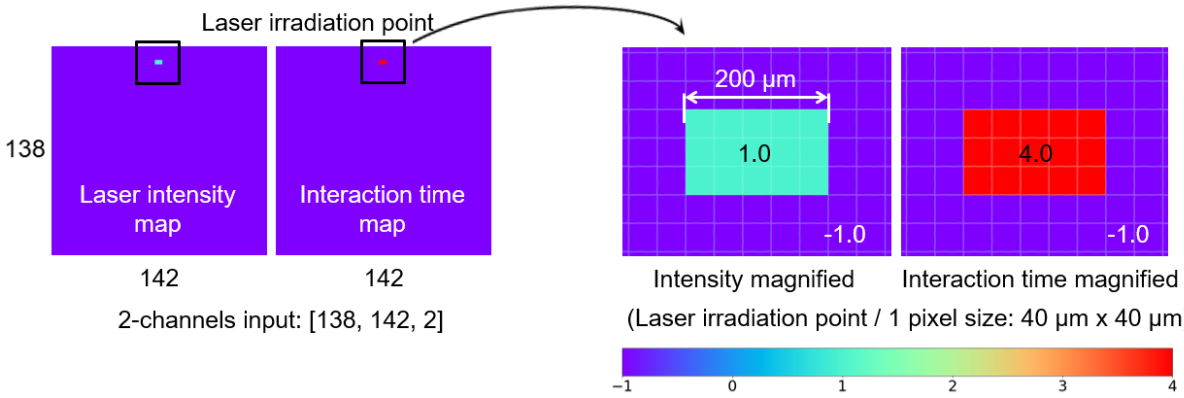

Figure 3.5. Two-channel input consisting of laser intensity and interaction time maps. The numbers inside the square bracket [138, 142, 2] are the number of pixels in height, width, and channel directions, respectively. Laser irradiation points in laser intensity and interaction time maps are magnified on the right. Shown process condition was 1800 W and 5.313 mm/s.

As presented in the magnified images, normalized I_0 and t_i values (1.0 and 4.0, respectively) were assigned at the laser irradiation point. The normalizations were calculated by I_0/I_0^* and t_i/t_i^*, respectively (I_0^* and t_i^* were respectively calculated with respect to P = 1800 W and v = 21.25 mm/s). Thus, the normalized intensities were 0.0625, 0.125, …, 1.0, and the normalized interaction times were 0.0625, 0.0884, …, 4.0, as shown in **Figure 3.4**. The normalized process condition was inserted at the laser irradiation point, which was a rectangle with 3 × 5 pixels (1-pixel size = 40 μm; 5 pixels in width = 200 μm = D_{beam}), and -1.0 was assigned at the other pixels. The number of pixels in height (three pixels) was determined by trial and error, near the filter size (four pixels). The two-channel image tensor of shape (H = 138, W = 142, C = 2) was input to the first generator.

To construct a robust AI model with a good prediction capability, the [(I_0, t_i), SEG] pairs in the training set for the first generator were augmented by adopting three different imaging techniques: flip left and right (× 2), image transformation (× 50), and horizontal translation (× 11). Therefore, for each training process condition, × 1100 augmentation was performed; as such, the total number of [(I_0, t_i), SEG] pairs in the training set for G_1 was 42900 (39 × 1100). Similarly, the [SEG, OM image] pairs in the training set for the second generator were augmented by flip left and right (× 2) and horizontal

translation (× 21); therefore, the total number of training data pairs was 1638 (39 × 42). A comprehensive data flowchart is given in **Figure 3.6**.

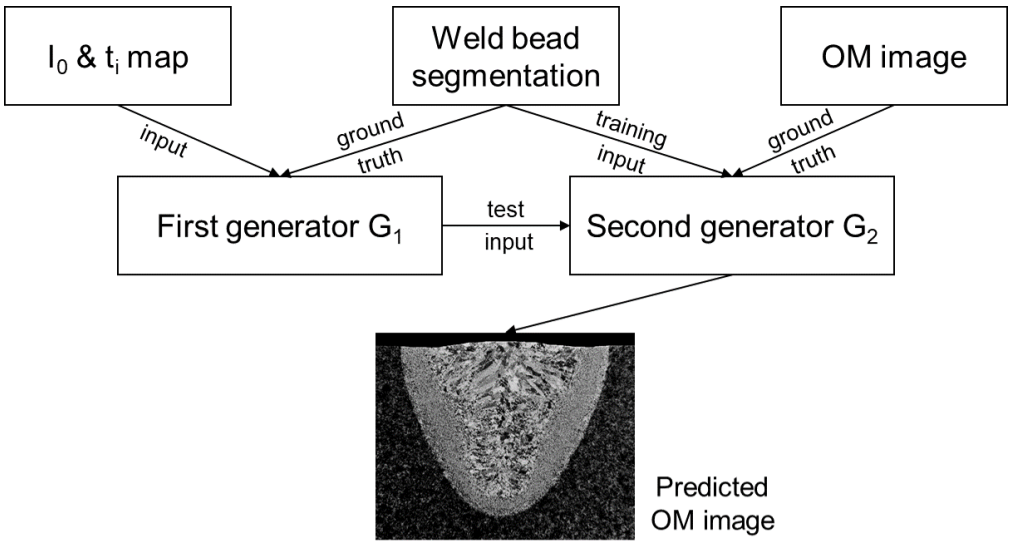

Figure 3.6. Data flowchart for the proposed deep learning model.

Note that the AI model must be trained with a large amount of data to cope with various situations, and thus proper data augmentation is essential for a successful learning. For instance, in laser keyhole welding, the keyhole interface fluctuates by high interfacial forces such as capillary and thermocapillary forces and recoil pressure [19, 20], so the bead shape is not always constant but varies a bit. Therefore, for the AI model to learn about the slightly different bead shapes caused from the multi-physics phenomena, the weld bead segmentation maps (ground truth) in the first generator were augmented by applying a random weak shear transformation. Meanwhile, corresponding inputs of I_0 and t_i maps were not transformed because the AI model had to learn that slightly changing bead shapes could occur from the same laser process condition.

3.3 Deep Learning Model

The deep learning model proposed in this study is composed of two training sessions of the first and second generators (G_1 and G_2) (**Figure 3.7**).

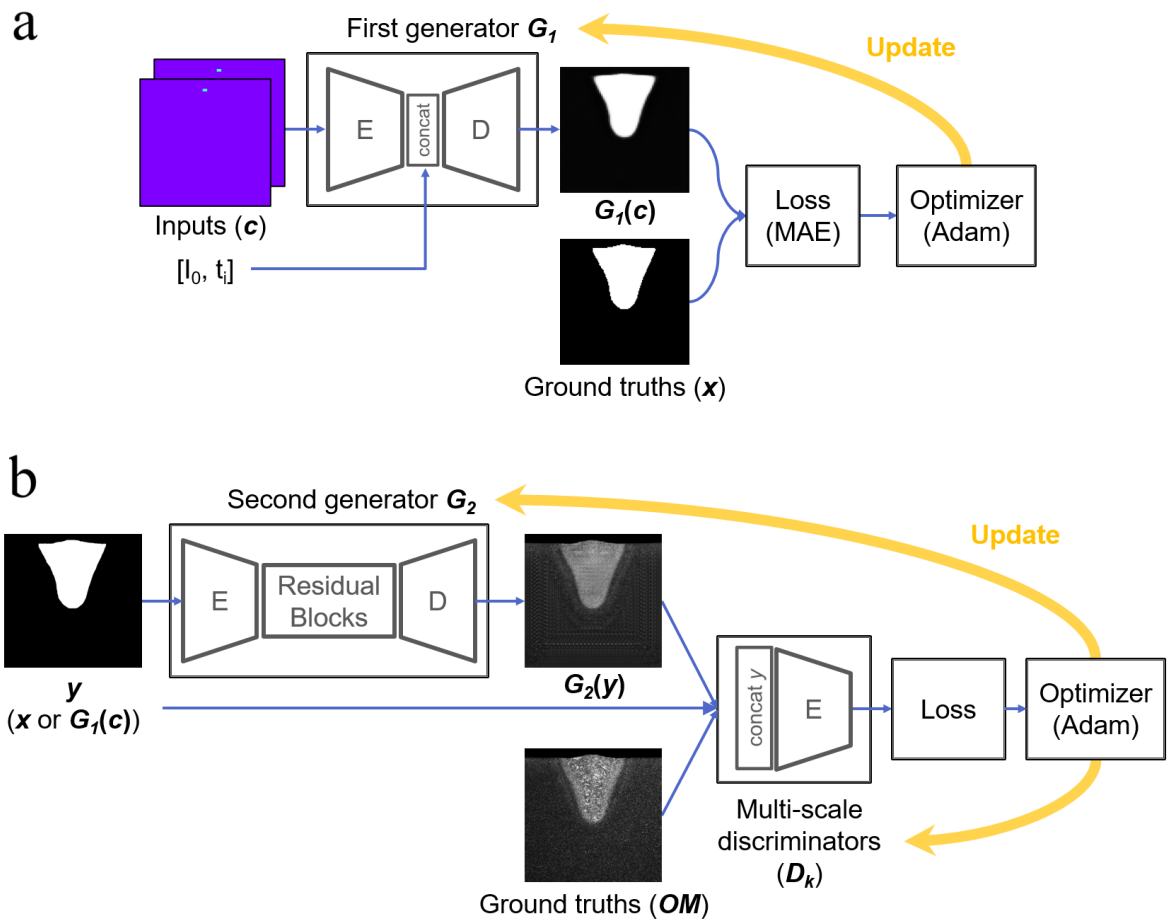

Figure 3.7. Flow diagram for the (a) first and (b) second generators. Shown process condition was 1800 W and 21.25 mm/s.

In the first generator, the two-channel input was translated into a weld bead segmentation map, through the CNN-based encoder–decoder network. Through the encoding line, the training filters in the CNN extract essential information contained in the input image according to the given loss function, and through the decoding line, the filters reconstruct the encoded features to the desired image. In the second generator, the weld bead segmentation map was input, and the generator learned to fill the map into an OM image, through the CNN-based cGAN network (pix2pixHD). Both deep learning models were implemented using Python™ and the TensorFlow™ library, on a i7-7820X CPU and dual GTX 1080 Ti GPUs.

3.3.1 The first generator

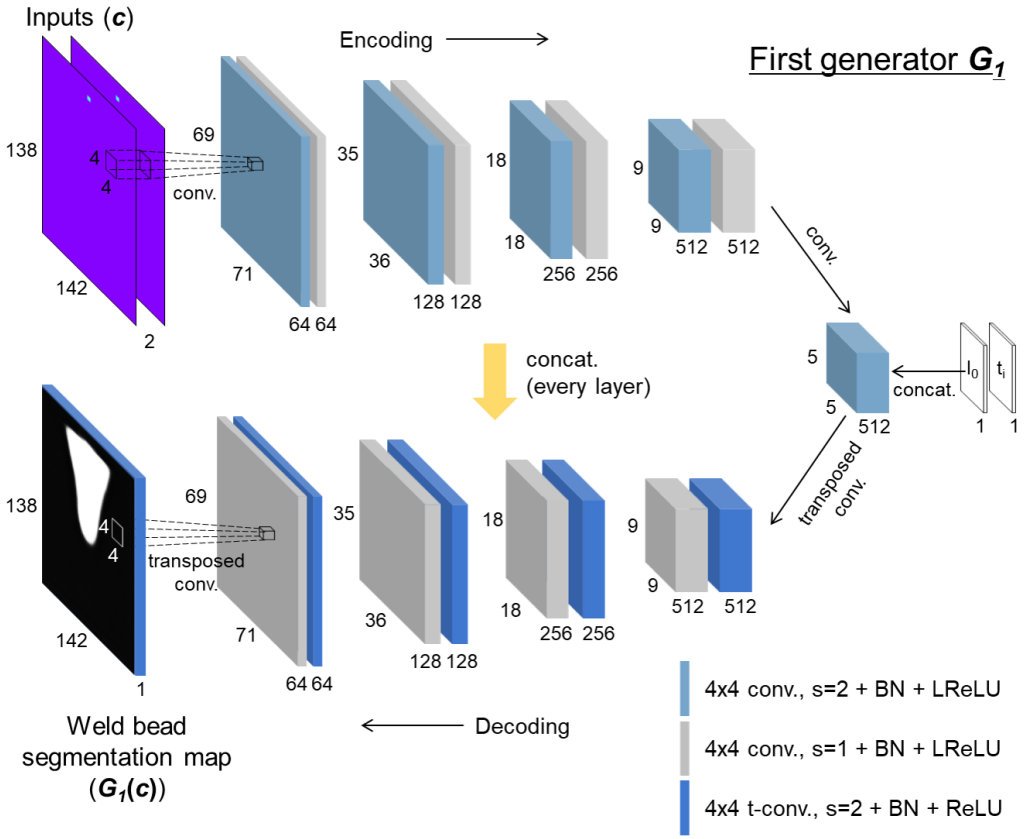

Figure 3.8. Detailed structure of the first generator. It consists of successive convolution-BN-activation layers.

A flow diagram and detailed structure of the first generator are presented in **Figure 3.7(a)** and **Figure 3.8**, respectively. As shown, the two-channel input and weld bead segmentation ground truth were denoted by c and x, respectively, and the predicted (generated) segmentation map from the generator was $G_1(c)$. Mathematically, the first generator was defined as

$$G_1 = G_1(c;\theta^{G_1}), \tag{3.1}$$

where θ^{G_1} are the training variables in G_1, *i.e.*, weights and biases in the convolutional processes, which are filters W and bias b in **Figure 1.14**. θ^{G_1} was trained to translate the two-channel input to a bead segmentation map, according to the following mean absolute error loss function J^{G_1} (mean of absolute difference between the ground truths and the generated segmentation images) [86]:

$$J^{G_1}(\theta^{G_1}) = \frac{1}{HW} \sum_{i=1}^{H=138} \sum_{j=1}^{W=142} \left| x - G_1(c) \right|. \quad (3.2)$$

The optimizer subsequently updates θ^{G_1} toward minimizing the given loss function by backpropagation, *i.e.*, to decrease the difference between the ground truths and the predictions. For the optimizer, an adaptive moment estimation (Adam) proposed by Kingma and Ba [65] was used.

In **Table 3.2**, the three operation blocks used in the first generator (**Figure 3.8**) are presented, with their configurations. These layers are repeated through several layers, and as the network deepens, higher levels of features included in the input image are extracted [58, 59]. It is noteworthy that the batch normalization was not adopted in the first light blue block and the last dark blue block represent operations of 4 × 4 transposed-conv. with stride 2–tanh, without BN.

Table 3.2. Operation blocks adopted in the first generator (s: strides) (**Figure 3.8**).

Blocks	Layers
Light blue	4 × 4 conv. (s=2)–BN [62]–LReLU [64] (max(0.2*x*, *x*))
Gray	4 × 4 conv. (s=1)–BN–LReLU (max(0.2*x*, *x*))
Dark blue	4 × 4 transposed-conv. (s=2) [57]–BN–ReLU [63] (max(0, *x*))

As shown in **Figure 3.8**, in the encoding line (top row), the two-channel input of the laser processing parameters was encoded to smaller dimensions through series of convolutional layers; additionally, through the decoding line (bottom row), the encoded bottleneck (H = 5, W = 5, C = 514) was upsampled to achieve input-level resolution (H = 138, W = 142, C = 1).

Additionally, skip connections between the encoder and decoder were used (orange vertical arrow in **Figure 3.8**), which were concatenation processes in each layer's channel direction (U-Net; [60]). In the bottleneck layer, the two laser processing parameters of I_0 and t_i were broadcasted and concatenated in the channel direction, to directly provide information about the process condition to the upsampling line. The concatenations were denoted by "concat." in the figure.

3.3.2 The second generator

After the first generator was trained, the second generator was trained with the weld bead segmentation map as an input and the OM image as a ground truth, as presented in **Figure 3.7(b)**. Both were (H = 552, W = 568, C = 1) in size, and in the case of such a high-resolution image generation problem, unstable generations have been a critical problem. For instance, when applying pix2pix [38] to our second generator, many repeated patterns were observed in the generated images. Hence, in this study, pix2pixHD [40] was employed in the second generator, which demonstrated particularly good results in a high-resolution image-to-image translation problem. Its primary framework is a CNN-based cGAN, with multi-scale generators and discriminators, and deep residual networks.

In **Figure 3.7(b)**, a data flow diagram is shown, with the second generator, multiscale discriminators, loss, and Adam optimizer. The input weld bead segmentation map is denoted by *y*, which was the result of the first generator ($G_1(c)$ in the validation and the test phases) or SEG (x, in the training phase). It is noteworthy that in the case of $G_1(c)$ as an input, it was 4× upsampled using bilinear interpolation before input to the second generator ((138, 142, 1)→(552, 568, 1)). The ground truth OM image was denoted by *OM*. Subsequently, the input y was translated to the OM image through the second generator, which was trained according to the loss functions given by the multiscale discriminators. The multiscale discriminators (D_k) operated in three different scales: D_1 at an original size, and D_2 and D_3 respectively at 2× and 4× downsampled sizes. According to pix2pixHD, to correctly distinguish between real and fake from high-resolution images, the discriminator must have a large receptive field (extremely deep layers or large convolutional filters); however, the training data would be overfitted and a larger memory would be required. Instead, they adopted multiscale discriminators while fixing the number of layers and the filter size (all the discriminators share the same network structure). Operating on the coarsest allows the generator to produce overall consistent images and operating on the finest scale leads the generator to generate elaborate details.

Mathematically, the second generator and the discriminator are expressed as

$$\begin{aligned} G_2(y) &= (G_{global}(y'), G_{local}(y)), \\ D_k &= D_k([G_2(y)\ or\ OM]\,|\,y) \end{aligned} \quad . \tag{3.3}$$

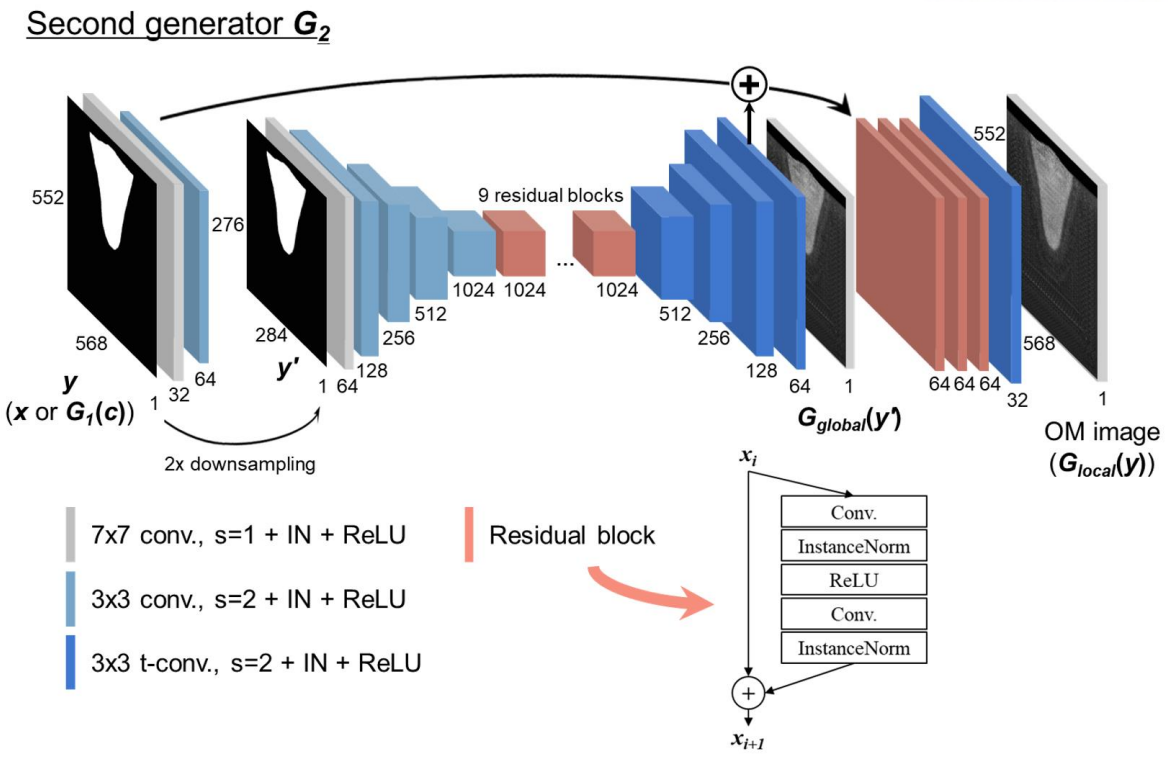

Figure 3.9. Detailed structure of the second generator with the residual block configuration.

As written, the second generator is a tuple of a global generator network (G_{global}, middle encoder-9 residual blocks-decoder network in **Figure 3.9**) and a local enhancer network (G_{local}, the remaining network excluding G_{global}, in **Figure 3.9**). Both networks were of an encoder-residual blocks-decoder structure but operated at different scales, *i.e.*, G_{global} at a 2× downsampled scale (H = 276, W = 284) and G_{local} at the full scale (H = 552, W = 568). *y'* denotes the 2× downsampled *y*, which was the input to the global generator. The vertical bar *y* represents the concatenation process in the channel direction between the weld bead image and *y*. Detailed configuration for the operation blocks in the second generator is presented in **Table 3.3**. Note that the last gray block in the decoders are operations of 7 × 7 convolution with stride 1–tanh.

Table 3.3. Operation blocks adopted in the second generator (s: strides) (**Figure 3.9**).

Blocks	Layers
Gray	7 × 7 conv. (s=1)–InstanceNorm (IN) [87]–ReLU
Light blue	3 × 3 conv. (s=2)–IN–ReLU
Dark blue	3 × 3 transposed-conv. (s=2)–IN–ReLU

The local enhancer network receives the last 64-channel feature map in the global generator (element-wise summation with the same-shaped encoded feature map) and raises its resolution through the following three residual blocks–decoder network. Training was performed in the order of global generator, local enhancer, and both. The configuration of the residual blocks [81] is shown in the figure. It is noteworthy that other well-known residual structures such as ResNet [80], full pre-activation [88], and EDSR [82] were tested, but did not achieve significantly better results. In addition, to reduce the computational cost, several simplified models were tested, such as removing the local enhancer (only global generator), decreasing the number of residual blocks, using only two discriminators, and utilizing skip connections [60] between the encoder and decoder; however, acceptable qualities could not be obtained.

In **Figure 3.10**, the discriminator structure is shown. It is the CNN-based encoder, and the light blue and gray blocks are operations of 4×4 convolution with stride 2–IN–LReLU and 4×4 convolution with strides 1–IN–LReLU, respectively.

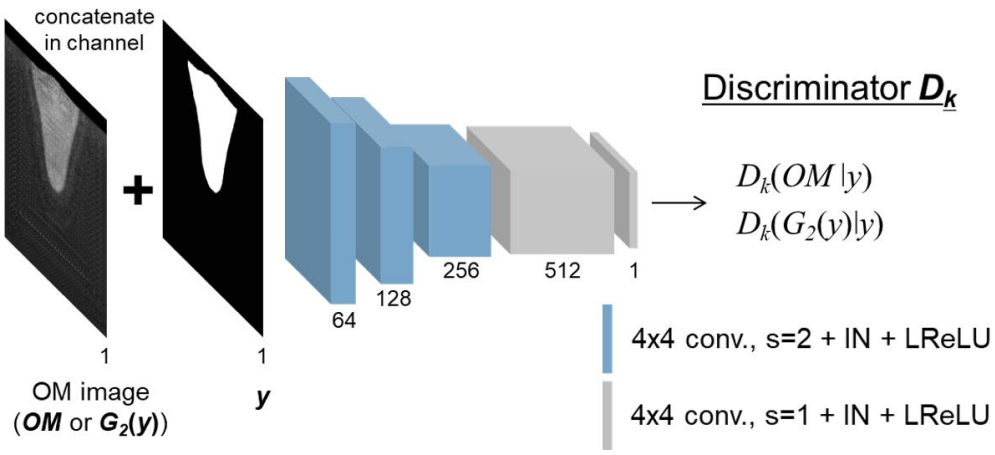

Figure 3.10. Detailed structure of the *k*-th discriminator.

For the loss function, based on the traditional GAN loss (J_{GAN}) that was given by ($\mathbb{E}$: expectation)

$$J_{GAN}(G_2, D_k) = \mathbb{E}_{OM,y}[\log D_k(OM|y)] + \mathbb{E}_y[\log(1 - D_k(G_2(y)|y))], \quad (3.4)$$

the full objective function was constructed with a feature matching loss (J_{FM}), as depicted in pix2pixHD:

$$\min_{G_2}\left(\left(\max_{D_1,D_2,D_3}\sum_{k=1,2,3}J_{GAN}(G_2,D_k)\right)+\lambda\sum_{k=1,2,3}J_{FM}(G_2,D_k)\right). \tag{3.5}$$

For an actual application of the cost, least squares GAN (LSGAN) [89] was adopted for a stable learning, and the loss function for the *k*-th discriminator is given by:

$$J^{D_k}=\frac{1}{2}\mathbb{E}_{OM,y\sim p_{data}(OM,y)}[(D_k(OM|y)-1)^2]+\frac{1}{2}\mathbb{E}_{y\sim p_y}[D_k(G_2(y)|y)^2], \tag{3.6}$$

where J^{D_k} is a cost function for D_k. The optimizer trains each discriminator's training variable by minimizing the given loss function, that is $D_k(OM|y)\to 1$ and $D_k(G_2|y)\to 0$. It implies that when the grount truth (*OM*) is input to the discriminator, it is trained to distinguish the input as the real (1), and the generated image ($G_2(y)$ *i.e.* $G_{global}(y')$ or $G_{local}(y)$) as the fake (0). The final loss function for the multi-scale discriminators is $J^D=J^{D_1}+J^{D_2}+J^{D_3}$. The loss function for the generator is

$$J_k^{G_2}=\frac{1}{2}\mathbb{E}_{y\sim p_y}[(D_k(G_2(y)|y)-1)^2], \tag{3.7}$$

where $J_k^{G_2}$ is a cost for G_2, given by the *k*-th discriminator. The optimizer updates generator's training variable such that $D_k(G_2|y)\to 1$ (adversarial with the discriminator's training). So as the training progresses, the generator creates more and more realistic images, to deceive the discriminator. Additionally, a feature matching loss was added to the loss function of G_2, which is

$$J_{FM,k}=\mathbb{E}_{OM,y\sim p_{data}(OM,y)}\sum_{l=1}^{L}\frac{1}{N_l}[\left\|D_k^{(l)}(OM|y)-D_k^{(l)}(G_2(y)|y)\right\|_1], \tag{3.8}$$

where $J_{FM,k}$ is the feature matching loss in the k-th discriminator, and l, L and N_l denote the index of layers in each discriminator ($D_k^{(l)}$: D_k's l-th layer), the total number of layers except for the last layer, and the number of elements in each layer, respectively (L=4: three light blue blocks and one gray block as shown in **Figure 3.10**). Matching the extracted features from the two kinds of inputs (OM and $G_2(y)$) helps generator produce more sophisticated images, especially in the high-resolution problems. The final loss function for the second generator is

$$J^{G_2} = J_1^{G_2} + J_2^{G_2} + J_3^{G_2} + \lambda\left(\frac{1}{K}\sum_{k=1}^{K} J_{FM,k}\right), \tag{3.8}$$

where K is the number of multi-scale discriminators (3) and λ is a coefficient (set to 10).

3.3.3 Training details

For the first generator, the learning rate, batch size, and exponential decay rate for the first moment estimate in the Adam optimizer (β_1) were 10^{-7}, 100, and 0.9, respectively. For the second generator, they were 0.0002 (linearly decaying to 1/100 from 0.0002 for every 100 epoch), 1, and 0.5, respectively. All the weights were initialized to a Gaussian distribution with a mean of 0.0 and a standard deviation of 0.02 [37], and all the bias to 0.0, in both G_1 and G_2. All the downsamplings were conducted by average pooling. The root mean square errors (RMSEs) of the first generator for the training and validation sets are given in **Figure 3.11** according to the number of epochs. As shown in the figure, the validation error continued increasing after reaching the minimum at epoch 295, so the training was stopped there. In the second generator, training was conducted as long as possible considering the computing time, and was stopped at epoch 160.

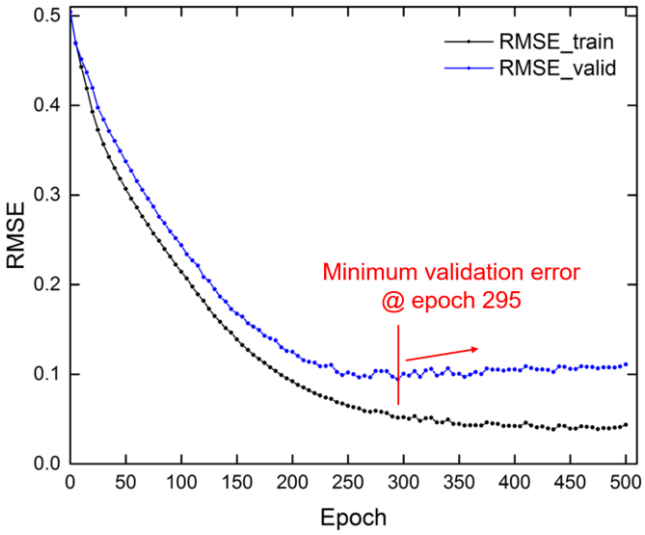

Figure 3.11. RMSE loss curves of the first generator for the training and validation datasets.

3.4 Results and Discussion

The prediction results for the test dataset are presented in **Figure 3.12**. In each row from top to bottom, the process conditions were A, B, C, and D (see **Figure 3.4**). In the first two columns, inputs (c) to the deep learning model (laser intensity map and interaction time map) are shown separately, with the normalized processing parameters (numbers inside the parenthesis are the corresponding laser powers or scanning speeds). In the third column, the prediction results of the weld bead segmentation ($G_1(c)$; output of the first generator at epoch 295) are drawn and the corresponding ground truths (x) are given in the fourth column. In the fifth column, final predictions of the OM image ($G_2(G_1(c))$; output of the local enhancer network when the epoch was 160) are presented, and their ground truths (OM) are given in the final column. It is noteworthy that in the test phase, the input to the second generator was the first generator's results ($G_1(c) \rightarrow y$), as explained previously.

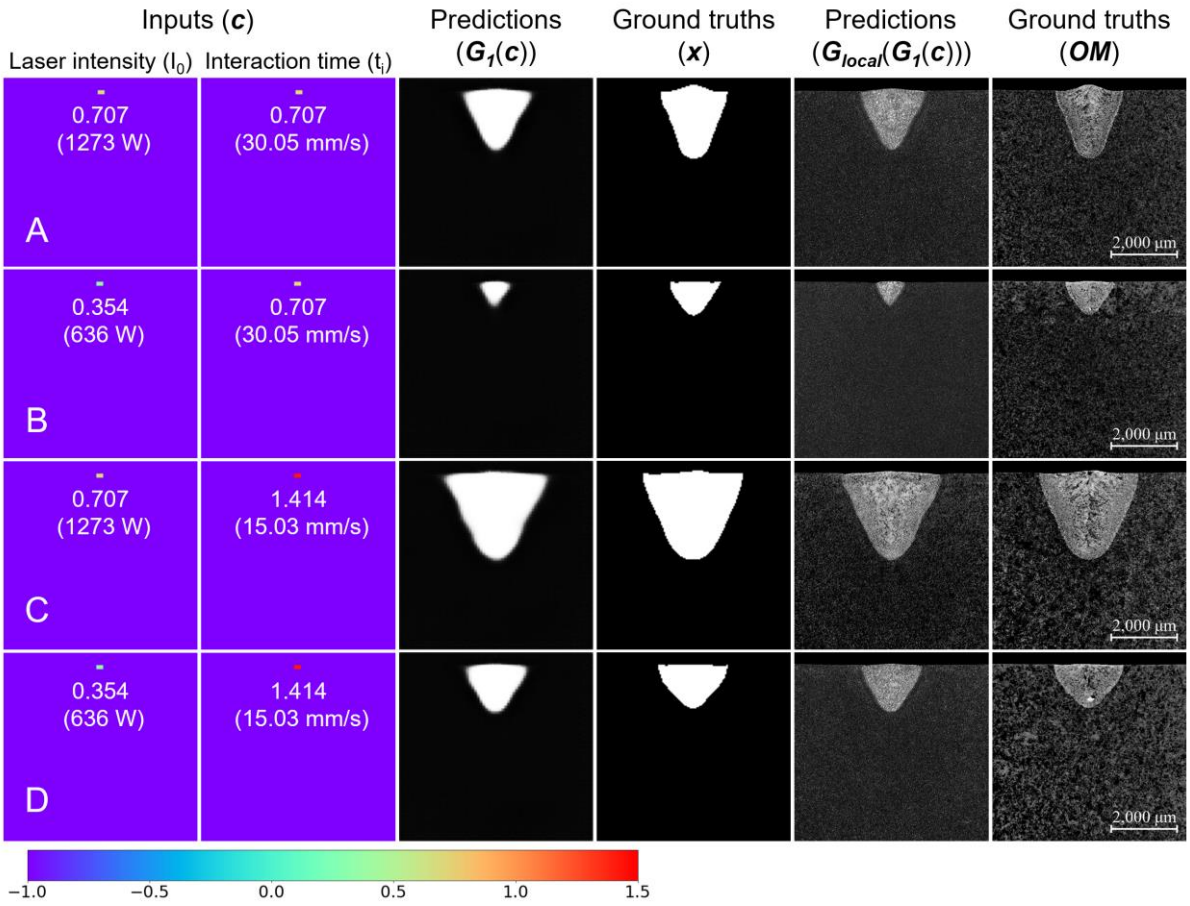

Figure 3.12. Test set results (from top to bottom, A, B, C, D). The first and the second columns are input normalized laser intensity and interaction time maps (c), respectively. The third and the fourth columns are prediction results for the first generator ($G_1(c)$) and their ground truths (x), respectively. In the fifth and the sixth columns, predicted OM images ($G_2(G_1(c))$) and ground truths (OM) are presented, respectively.

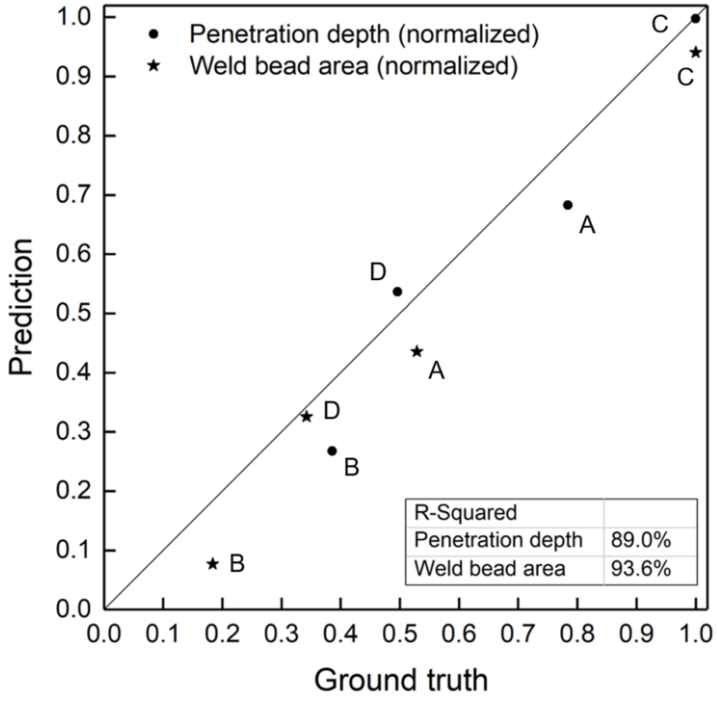

Figure 3.13. Graphical comparison between the prediction result (*y*-axis) and the ground truth (*x*-axis) for four test process conditions (A, B, C, D), in terms of penetration depth (marked by circle) and weld bead area (marked by star). All the values were normalized with respect to the maximums. Corresponding R-Squared accuracies are written in the bottom-right corner.

As shown in **Figure 3.12**, the AI model created the weld bead images that is quite similar to the actual cases, in terms of bead shape and penetration depth. To evaluate the performance of the model, the penetration depth and weld bead area of the prediction result and ground truth for four test process conditions (A, B, C, and D) were measured, and shown in **Figure 3.13**. All the data of penetration depth and bead area in the figure were normalized with respect to the maximum penetration depth and maximum bead area, respectively. The *y*-axis and *x*-axis were prediction result and ground truth, respectively, and the penetration depth was marked by circle and the weld bead area by star. As shown in the figure, the data distribution appears to be close to the $y=x$ line (prediction ≈ ground truth) and calculated R-Squared accuracy was 89.0% for penetration depth and 93.6% for weld bead area. In terms of the R-Squared accuracies and overall aspect of the weld bead in **Figure 3.12**, it can be concluded that the AI model predicted the weld beads fairly well, from the laser processing parameters.

Figure 3.14 presents predicted and ground truth OM images of the process condition C that demonstrated the largest bead size and the highest bead shape accuracy (**Figure 3.13**; closest to the $y=x$ line). Even though the predicted grain details were not fully actualized, the shape and size of the overall weld bead were well predicted, and the molten zone near the bead boundary was fairly well

distinguished from the central region. Furthermore, despite the high resolution of the image, the generated images were clear without any unstable or repeated regions, owing to the multi-scale generators and discriminators, and deep residual networks adopted in the second generator [41, 82-85].

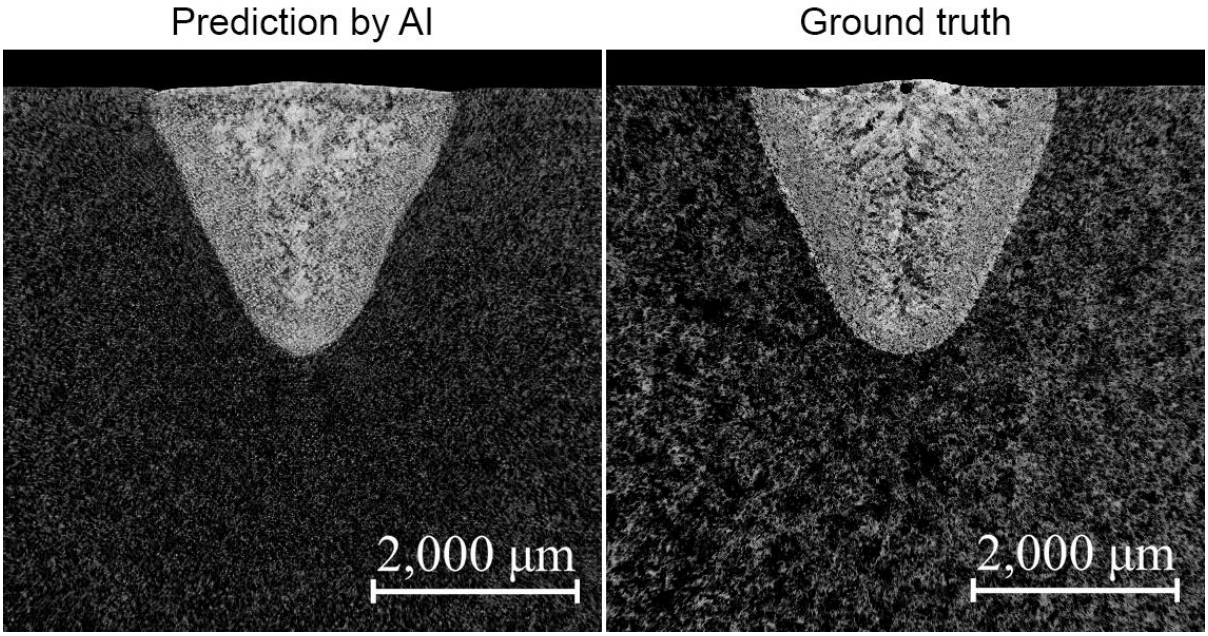

Figure 3.14. (Left) High-resolution (H = 552, W = 568, C = 1) OM image predicted by AI. (Right) Ground truth image. The laser process conditions were 1273 W and 15.03 mm/s (C).

In **Figure 3.15**, **Figure 3.16**, and **Figure 3.17**, the insides of the two generators were open to track the predicting process when the process condition of 1800 W, 5.313 mm/s (training set; **Figure 3.15**), 1273 W, 60.1 mm/s (validation set, V1; **Figure 3.16**) and 1273 W, 15.03 mm/s (test set, C; **Figure 3.17**) were input. The top two rows are the first generator at epoch 295 and the bottom three rows are the second generator at epoch 160. Only the first channel (C = 1) was shown for brevity. In the first generator, the encodings of the input progressed along the top line from left to right and the decodings were conducted in the bottom line from right to left. In the second generator, the global generator was presented in the upper two rows (encodings in the first row and decodings in the second row), and the local enhancer was in the last bottom row. As presented, the local enhancer accepts the lower resolution feature maps from the global generator and increases the resolution to the full scale, as clearly shown by comparing two starred feature maps, which were the closest layers to each end. It is worth noting here that in this way, by adding extra enhancers operating at larger scales, it is possible to generate even higher resolution bead images [40] (H = 552, W = 568 → H = 1104, W = 1136 → …; although it will require more and more GPU cost), which also can help actualize the microstructures.

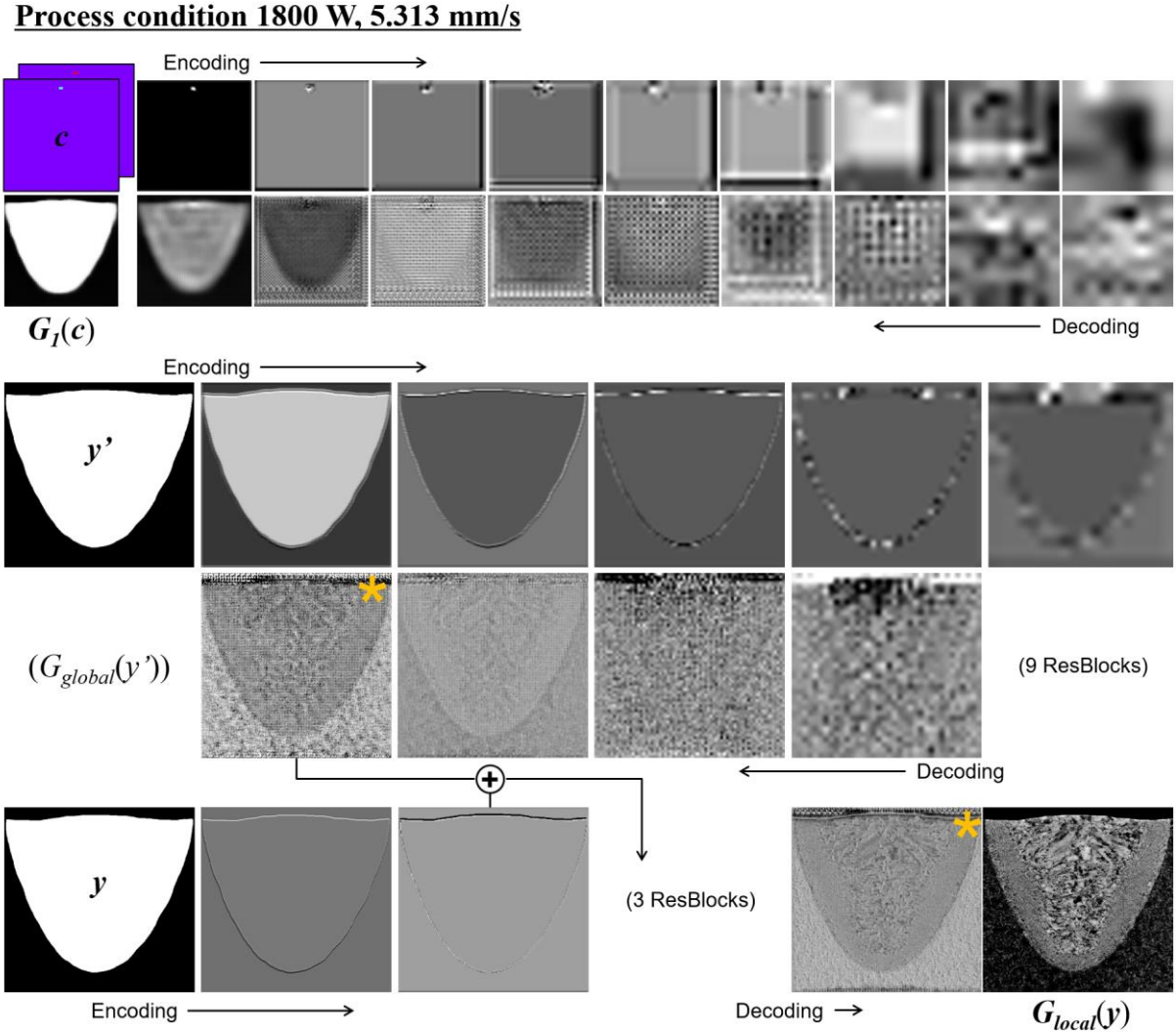

Figure 3.15. Insides of the two generators when the process condition of 1800 W and 5.313 mm/s was input (training set). Upper two rows are the first generator and lower three rows are the second generator. In the first generator, encoding progressed in the first row (from left to right) and decoding is conducted in the second row (from right to left). In the second generator, global generator (upper two rows) and local enhancer (last row) are shown with data pipelines.

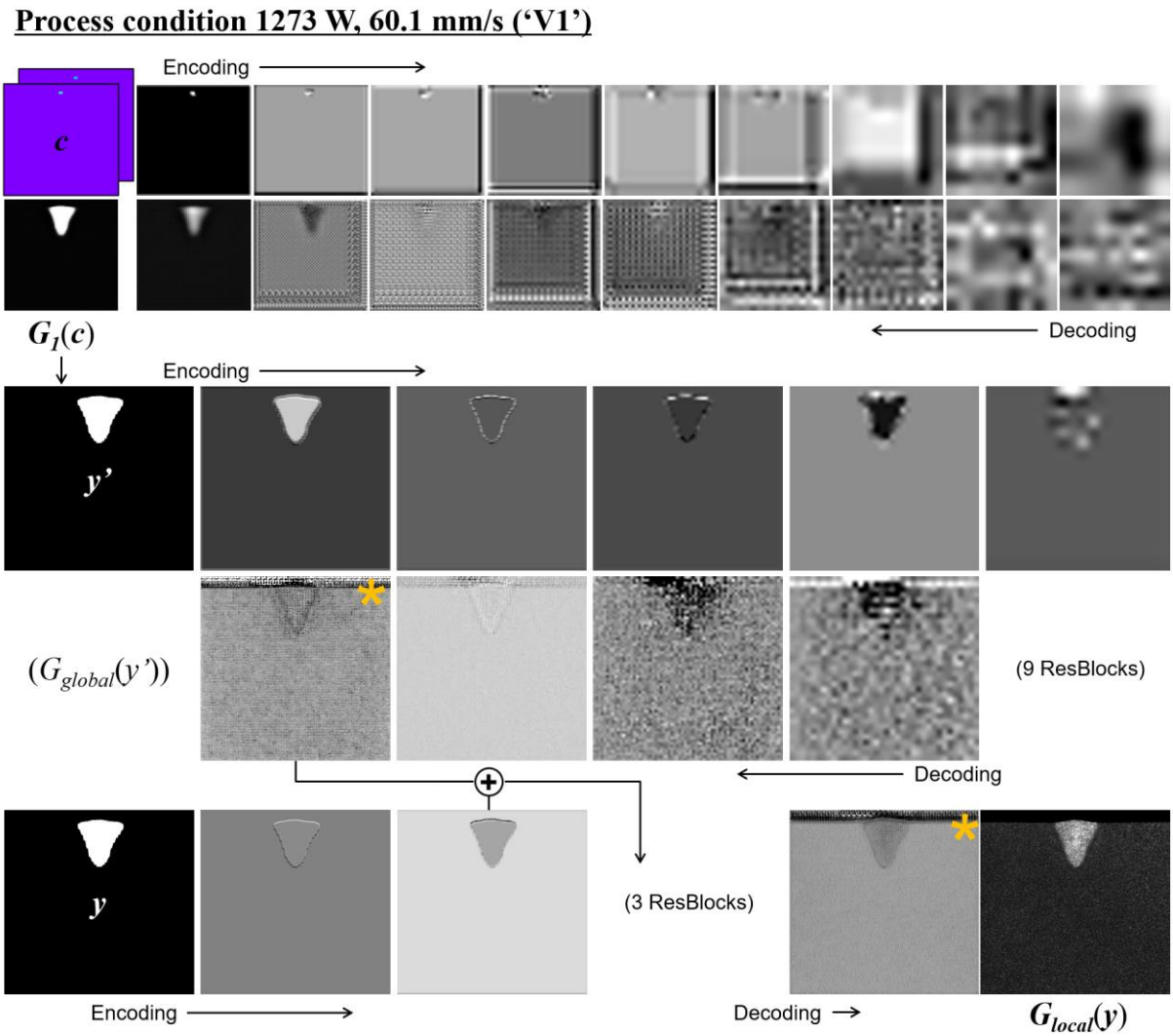

Figure 3.16. Insides of the two generators when the process conditions of 1273 W and 60.1 mm/s was input (validation set, V1). Upper two rows are the first generator and lower three rows are the second generator. In the first generator, encoding progressed in the first row (from left to right) and decoding is conducted in the second row (from right to left). In the second generator, global generator (upper two rows) and local enhancer (last row) are shown with data pipelines.

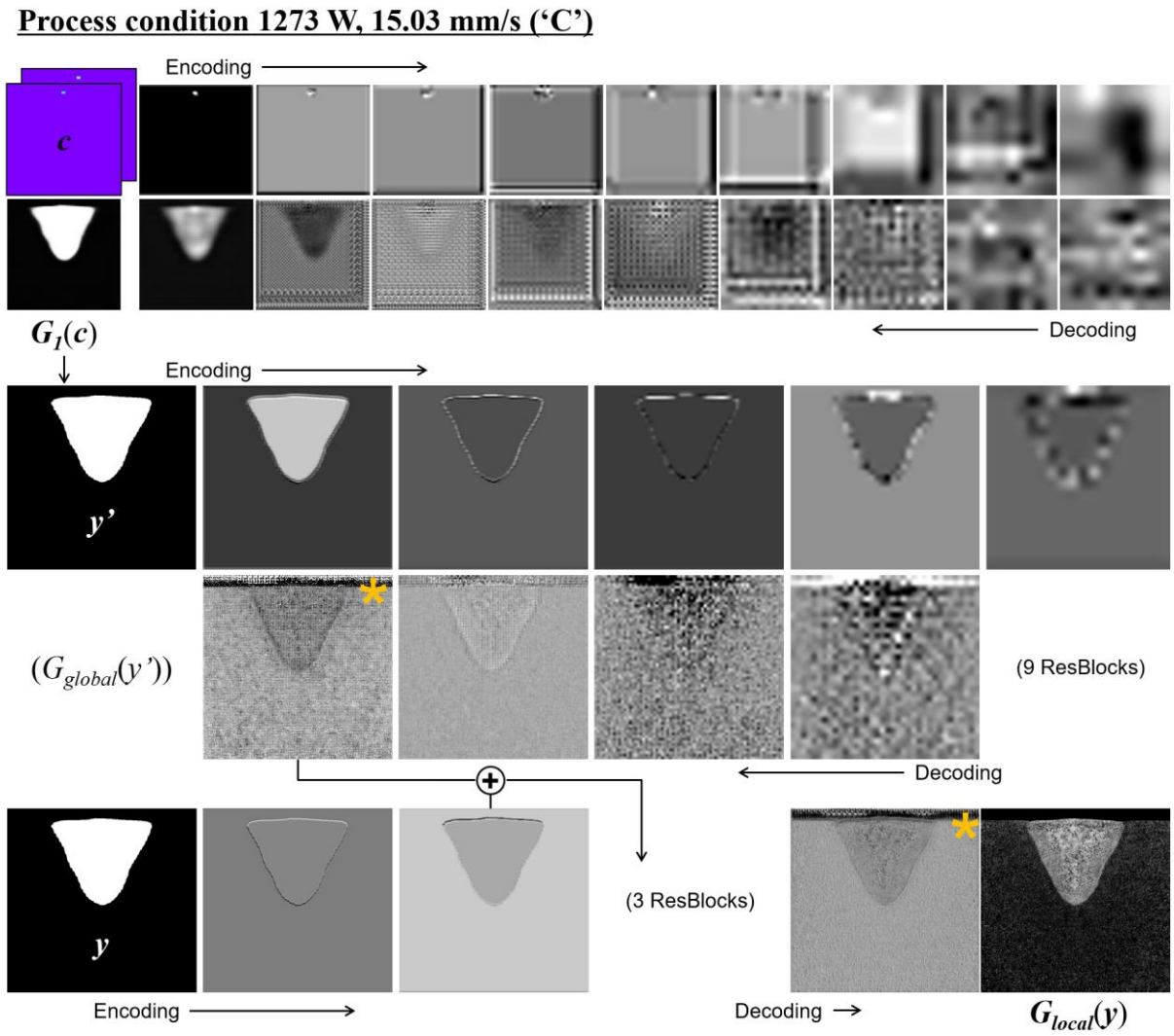

Figure 3.17. Insides of the two generators when the process conditions of 1273 W and 15.03 mm/s was input (test set, C). Upper two rows are the first generator and lower three rows are the second generator. In the first generator, encoding progressed in the first row (from left to right) and decoding is conducted in the second row (from right to left). In the second generator, global generator (upper two rows) and local enhancer (last row) are shown with data pipelines.

Figure 3.18 presents the extracted feature maps of the closest layer to the end of the local enhancer. The process conditions were 1800 W and 5.313 mm/s (training set) in the left figure, and 1273 W and 15.03 mm/s (test set, C) in the right figure. (They are the magnified second starred image in **Figure 3.15** and the image at the same layer in **Figure 3.17**, respectively). As shown, the heat affected zones (marked by yellow arrows in the figure) of the test set were fairly well recognized, as in the training set case. Meanwhile, the grain details inside the beads were realized but not correctly, as discussed in **Figure 3.14**. This remains as future work.

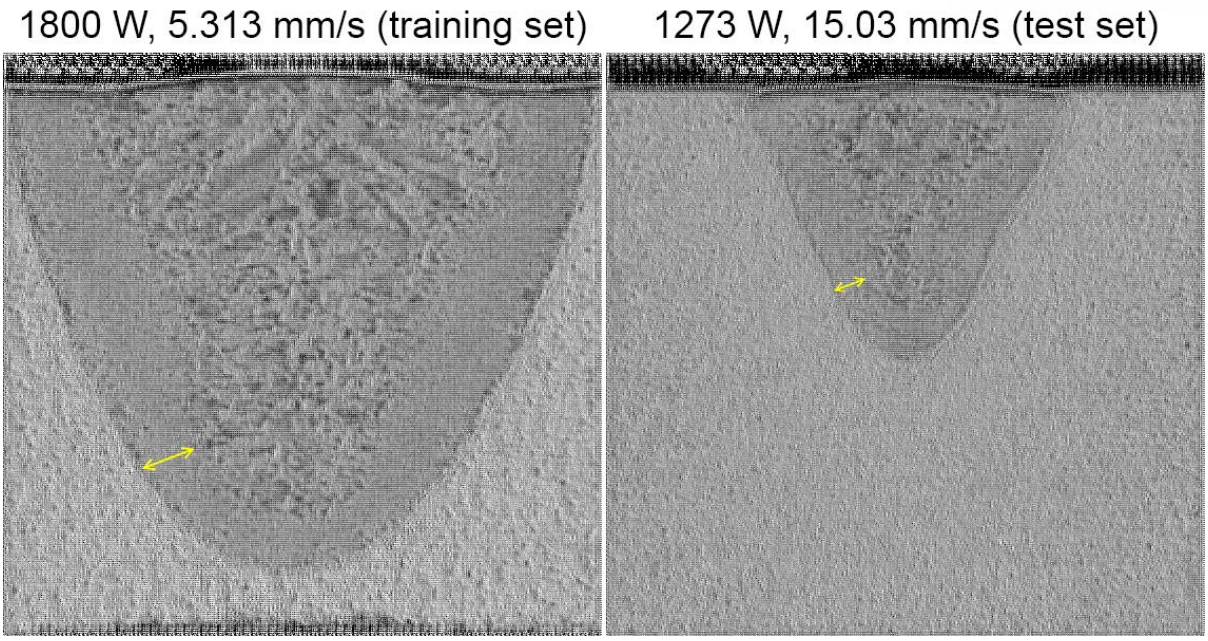

Figure 3.18. Feature map comparison between the training set (1800 W and 5.313 mm/s, left figure) and test set (1273 W and 15.03 mm/s, right figure). The feature maps were extracted from the last second layer in the local enhancer network.

3.5 Conclusion

A novel framework of predicting cross-sectional laser weld bead images was presented in this chapter, using state-of-the-art deep learning architectures. The first generator could predict the bead segmentation map (*i.e.* weld bead shape) from the two laser processing parameters of laser intensity and interaction time, and the second generator could fill the segmentation map into the optical microscopic bead image. Trained from only 39 pairs of raw data (39 laser process conditions), the generators predicted the laser weld bead with R-Squared accuracies of 89.0% for penetration depth and 93.6% for weld bead area, and the generated bead images were of high resolution without partial repetitions or distortions. Also, heat affected zone inside the bead was well distinguished from the keyhole area.

However, grain details inside the weld bead were not fully actualized, and the resolution of the generated bead image was less than that of the actual microscopic images (generally 500~2000 pixels on each side). Thus, future work includes scaling-up of the generated images in both generators which also can help realize the grain details (*e.g.* by stacking additional enhancer networks).

Chapter 4. Deep Learning Study on Self-Piercing Riveting

This chapter includes the published contents:

4.1 Introduction

Self-piercing riveting (SPR) is one of the most promising techniques for joining sheet materials. It has been drawing increasing attention especially in the automotive industry, as it can join similar or dissimilar sheets such as steel–steel, steel–aluminum, and even composite–metal combination which is difficult to process with conventional mechanical machining [90, 91]. A rivet pressed by a punch directly penetrates the top sheet, and the rivet legs at the bottom sheet are flared to fit the die shape and form a mechanical interlock. Using this technique, two or more sheets can be easily joined in a single-step without a preprocessed hole or thermal deformation.

To successfully perform the technique, the actual application of SPR must be preceded by the simulation of the process because the cross-sectional shape after the riveting determines quality of the joint [26]. Porcaro *et al.* [92] simulated the SPR of two aluminum alloy plates based on a two-dimensional axisymmetric model using the finite element method (FEM), and the simulation results for the deformed cross-sectional shape were compared with the experimental data. In addition to the geometrical modeling of the joining process, Casalino *et al.* [93] modeled a physical failure mechanism of the joint in the SPR process for aluminum alloy by solving governing equations using FEM, and experimentally validating their prediction results. Applying the failure investigation results from [93], Atzeni *et al.* [94] performed the SPR of aluminum sheets experimentally and numerically by FEM, and the deformed shapes were compared. Huang *et al.* [95] conducted FEM-based numerical simulation of SPR of aluminum alloy and steel sheets with various fracture criteria, and experimentally verified the model in terms of the deformed shape, force-displacement curve, and residual stress profile. Developing the previous FEM simulations, Carandente *et al.* [96] proposed an improved thermo-mechanical FEM model for various SPR processes, including thermal softening and strain hardening, and conducted experiments on SPR of aluminum sheets to verify the model. The predicted cross-sectional shape was compared with the experimental results. Ma *et al.* [97] numerically investigated the friction SPR of aluminum and magnesium using a three-dimensional thermomechanical-coupled FEM code, and the predicted deformed shape was compared with the experimental result. Hönsch *et al.* [98] presented the

FEM simulation of the SPR of aluminum alloys, and three joint shapes were predicted and compared with the experimental results. Without the FEM simulation, Haque *et al.* [99] proposed a rivet flaring model for predicting the amount of rivet flection from the force-displacement curves and geometries of the sheets, die, and rivet. For the SPR of carbon fiber reinforced plastic (CFRP), Di Franco *et al.* [90] performed experiments and FEM simulation for SPR between CFRP and aluminum alloy sheets, and the deformed shape was predicted. Also, Kroll *et al.* [100] experimentally performed the SPR of CFRP and aluminum alloy sheets by focusing on the fiber orientations, and simulated the joint formation by using FEM. Hirsch *et al.* [101] simulated the SPR of fiber reinforced polymers and metal sheets using the FEM. The simulation was based on a damage model to consider the deformation and failure behavior of the composites. As stated, most studies were based on FEM simulation, which gives insight into the process but typically requires careful selection of the mesh size and time step and consumes extensive computation time.

In this study, a novel deep learning framework was presented to predict the cross-sectional shape of the SPR joint from the scalar input punch force. Using the proposed deep learning architecture, we were able to obtain cross-sectional shapes including the location and deformed shape of the rivet and sheets, which settle the quality of SPR joint. At first, a concept of material segmentation was proposed, which classifies the cross-section by material so that can characterize the joint shape. To automatize this process from the cross-sectional optical microscopic images, we trained a convolutional CNN [35]-based encoder-decoder [79] structured deep learning model (DeepLabv3+; [102]). The next deep learning model was designed to predict the cross-sectional shape (material segmentation map) from an input punch force (a scalar value). It was a novel generative model based on the CNN, cGAN [36, 52], and residual blocks [80, 81], where predictive segmentation images were generated from a scalar value (namely, a scalar-to-seg generator). Note that the second deep learning architecture was the major predictive model, and the first segmentation model was employed to automatically supply the training data to the second model.

For a data preparation, SPR experiments were conducted with two kinds of sheet combination, which were CFRP and galvanized dual phase steel (GA590DP) sheets and steel alloy (SPFC590DP) and aluminum alloy (Al5052) sheets, by varying the punch force while fixing other process parameters. Using the observed cross-sectional OM images, the AI models were trained and validated. The segmentation model, which was trained with the SPR OM image data, segmented the test set OM images with 98.50% mean intersection-over-union (mIOU) and 99.78% mean pixel accuracy (mPA). Note that our segmentation model can segment the SPR OM images regardless of the material combination types (*i.e.* one trained model can segment both CFRP-GA590DP and SPFC590DP-Al5052 combinations, with the given accuracies). For the cross-section predictive model, which was trained with the material segmentation map, the prediction quality was assessed based on the foremost

geometric parameters that determine the joint quality, which were rivet head height (95.66%, 92.76%), interlock length (90.78%, 97.55%), and bottom thickness (96.16%, 98.60%) (the former: CFRP-GA590DP, the latter: SPFC590DP-Al5052). Mean accuracies were 94.20% and 96.31%, respectively, for the two material combinations. In summary, the key contributions in this study are as follows.

- We developed a novel deep learning framework for predicting cross-sectional shape in self-piercing riveting.

- We introduced the concept of material segmentation, to characterize the cross-sectional shape of rivet and sheets. Correspondingly, a state-of-the-art deep learning model of image segmentation was introduced along with appropriate hyperparameters.

- We proposed a novel scalar-to-segmentation generator, by modifying the conventional conditional GAN structure. A simple case study was performed to support the designed generator.

4.2 Data Preparation

In SPR, there are several process parameters which synthetically affect the joint quality, such as geometrical shape and dimension of rivet, sheets, and die, material properties, punch force and speed, etc [26]. The parameters and corresponding values used in this study are given in **Figure 4.1** and **Table 4.1**.

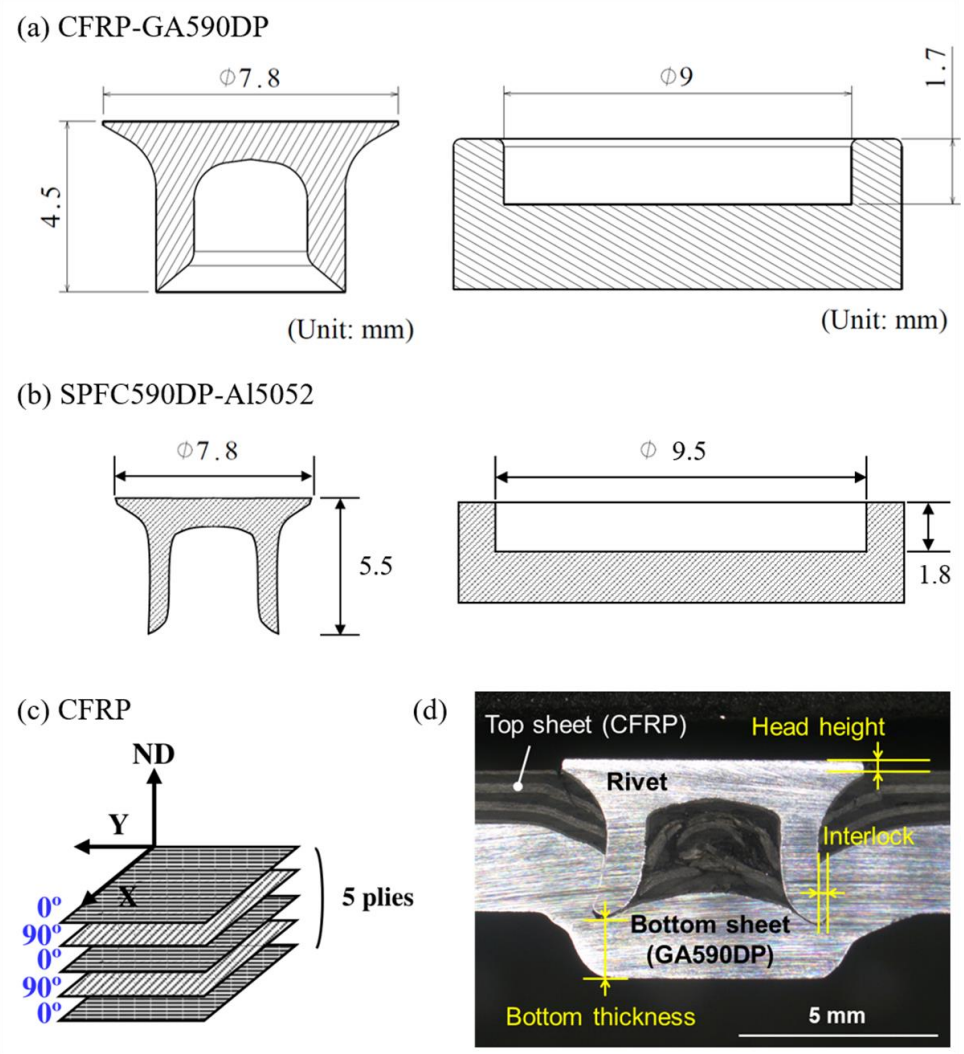

Figure 4.1. Geometrical dimensions of the rivet and die for (a) CFRP-GA590DP and (b) SPFC590DP-Al5052. Laminate sequence of cross-ply composites (0°/90°) and geometrical parameters of the SPR joint are shown in (c) and (d), respectively.

Two different types of material combination were used in the SPR experiment, which were thermoset CFRP as the top sheet and GA590DP steel as the bottom sheet, and SPFC590DP steel alloy as the top sheet and Al5052-H32 alloy as the bottom sheet. No adhesive or lubricant was used. The

CFRP consisted of 5 laminates with cross-ply ($0^o/90^o$), as shown in **Figure 4.1(c)**. The tensile strength, thickness, coating, and applied punch forces are listed in **Table 4.1** with the rivet and blank holder properties, and geometrical shape and dimension of the rivet and die are shown in **Figure 4.1(a)** (CFRP-GA590DP) and **Figure 4.1(b)** (SPFC590DP-Al5052). As presented in both figures, the die (supplied by BÖLLHOFF) had a basic flat-bottom shape. SPR joints were fabricated by a hydraulic riveting machine (Rivset Gen2, BÖLLHOFF) with a maximum setting force of 78 kN. Note that the punch speed was not varied (it was near 0.13 m/s), since the riveting machine was the hydraulic type.

Table 4.1. SPR process parameters used in the study.

Sheet materials	Location	Tensile strength (MPa)	Thickness (mm)	Coating	Punch force
CFRP	Top	1094 (Y-dir.)	1.3	None	29-70 kN
GA590DP	Bottom	610	2.3	Zn (43.8 g/m^2)	
SPFC590DP	Top	609	1.4	None	28-46 kN
Al5052-H32	Bottom	228	2.0	None	
Rivet	Hardness (Hv)	Coating	Blank holder (D: diameter)		Edge distance (mm)
Boron steel (supplied by BÖLLHOFF)	480 ± 30	Almac® (Al/Zn)	D_{outer}: 13 mm D_{inner}: 8.5 mm Pressure: 52.64 MPa		15

To assess the prediction quality, three geometrical indexes of the SPR joint were measured, namely, head height (distance between the rivet head surface and upper surface of the top sheet), interlock width (distance between the tip of the deformed rivet shank and the pierced point of the top sheet), and bottom thickness (remaining thickness of the bottom sheet after the riveting), as presented in **Figure 4.1(d)**. All geometrical indexes were measured twice for every OM image (left and right) and then averaged.

For the CFRP-GA590DP, the experiment was conducted with nine different punch forces (29, 34, 39, 44, 49.8, 55, 60, 65, and 70 kN) and repeated three times for each punch force. For the SPFC590DP-Al5052, seven punch forces (28, 31, 34, 37, 40, 43, and 46 kN) and repeated two times for each punch force.

4.3 Deep Learning Model

The concept of image segmentation in computer vision was adopted to characterize the cross-sectional shape of the self-pierced specimen from the OM image. Particularly, the segmentation was conducted material by material (rivet, upper sheet, and lower sheet) (**Figure 4.2**).

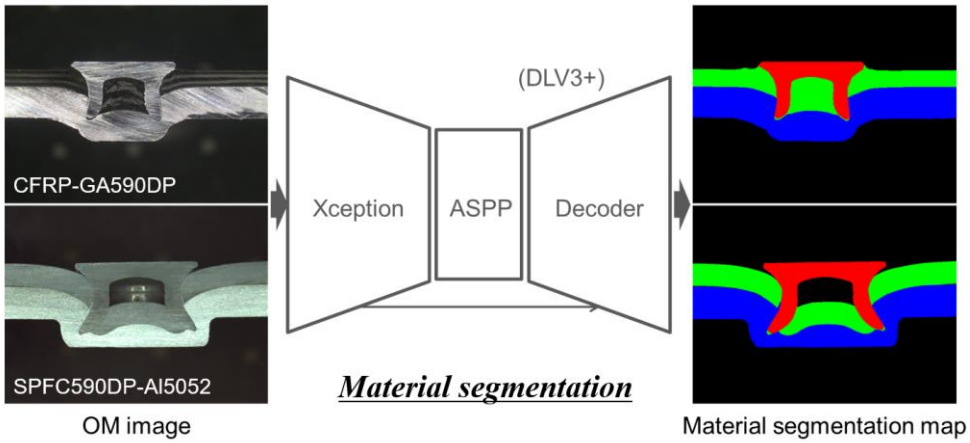

Figure 4.2. Overview of the material segmentation deep learning model.

Therefore, the process was named "material segmentation" in this study. In the segmenting process, unnecessary information contained in the OM image, such as surface details, the texture of the materials, image contrast, and noise, was removed, and only essential information such as geometry and location of the materials was retained. The segmented maps were then used as training data in the predictive model instead of the OM images, as shown in **Figure 4.3**. In this study, the predictive deep learning model was the major model, and the segmentation model was the ancillary model, which facilitated data supply to the predictive model (considering the need for further training in the future).

Google's DeepLabv3+ [102] was employed for the segmentation model, and for the predictive model, a generative model based on the CNN and cGAN architectures with residual blocks was adopted with a modified objective function. All the deep learning source codes were implemented using Python™ and TensorFlow™. The deep learning segmentation model is introduced in **chapter 4.3.1**, deep learning predictive model is introduced in **chapter 4.3.2**, and training information is presented in **chapter 4.3.3**.

Prediction of material segmentation map (scalar-to-seg generator)

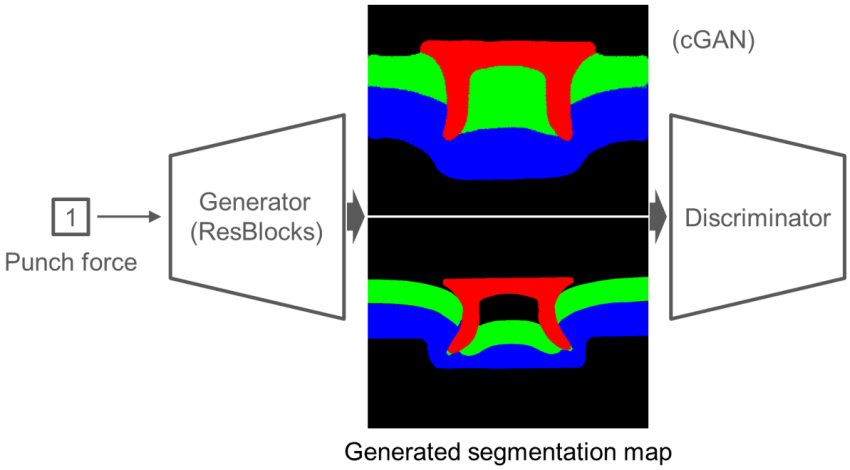

Figure 4.3. Overview of cross-sectional shape predictive deep learning model.

4.3.1 Material segmentation

To prepare the training data from the experimentally obtained cross-sectional OM images, pixel-by-pixel material segmentation was performed manually, including assigning different labels (colors) to different materials. For instance, in the material segmentation map shown in **Figure 4.2**, the red, green, blue, and black labels represent the rivet, upper sheet, lower sheet, and background, respectively. The OM image and its segmentation map were used as the input and ground truth, respectively, in the deep –learning segmentation model (DeepLabv3+). In **Figure 4.4**, a schematic architecture of the model is presented; the figure legend shows the six kinds of operation blocks.

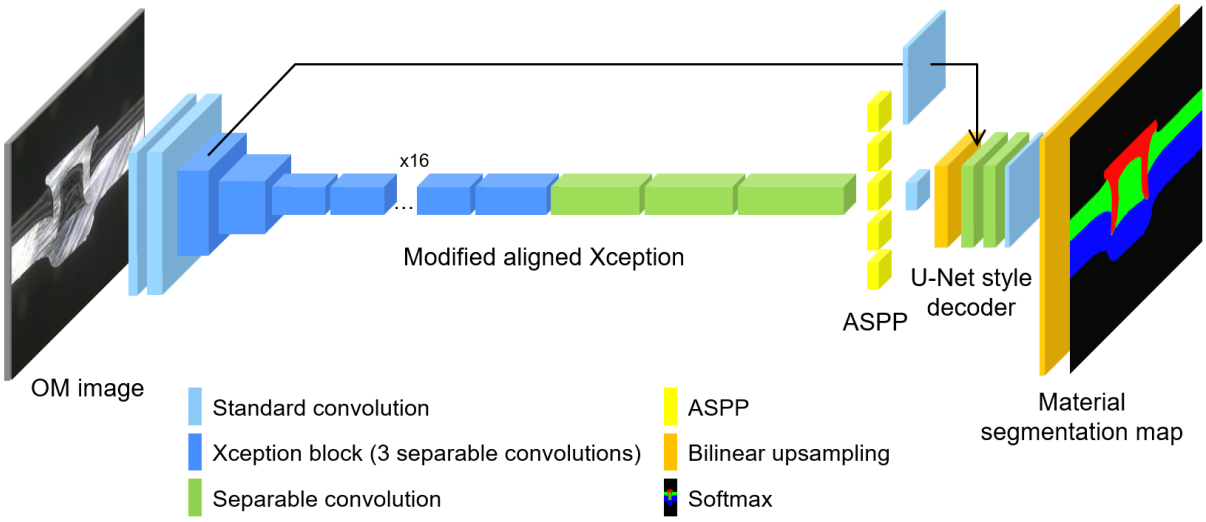

Figure 4.4. Architecture of the segmentation deep learning model. The structure was a CNN-based encoder-decoder with atrous separable convolutions.

As seen in the figure, a CNN-based encoder-decoder structure was used, where the input SPR OM image was encoded through modified aligned Xception and atrous spatial pyramid pooling modules, and the encoded features were reconstructed into the segmentation map through a U-Net style decoder. In this model, in line with the purpose of image segmentation, a separable convolution that effectively reduces the amount of computation and an atrous convolution that allows the weight filter to process a larger area with the same amount of computation were applied throughout the structure. In addition, Xception [103], which is a powerful deep learning architecture for image classification, was modified and applied in the encoding line, and the U-Net [60] structure, which concatenates on the encoded features to the decoding line, was utilized. The detailed explanation of the model is not presented in this dissertation for brevity; this information can be obtained from the DeepLabv3+ article [102].

4.3.2 Cross-sectional shape prediction

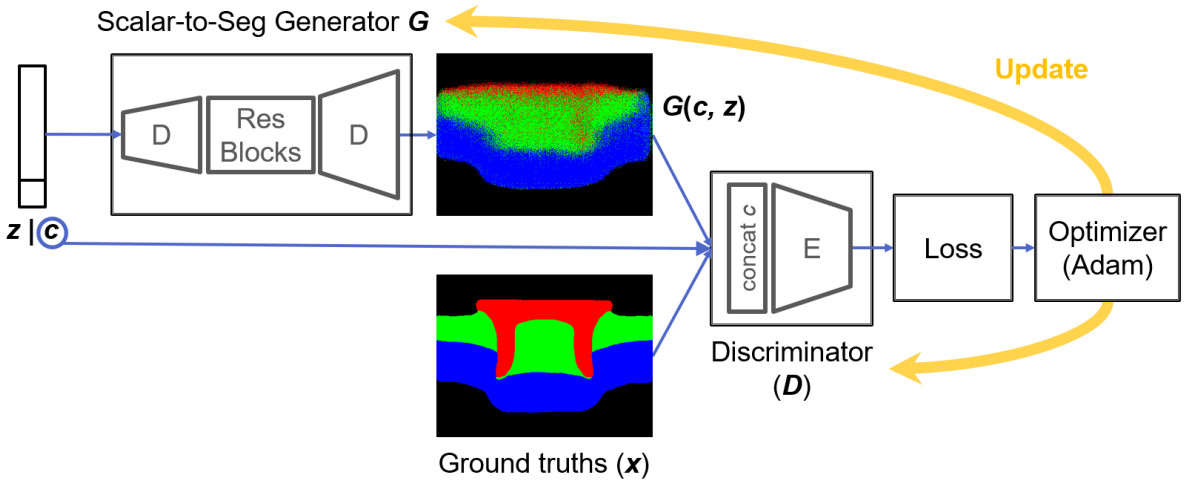

Figure 4.5. Flow diagram for the scalar-to-segmentation generator.

For the deep learning predictive model, a generative model based on the CNN and cGAN architectures with residual blocks was designed to predict the material segmentation map from a scalar punch force (namely, scalar-to-seg generator), as shown in **Figure 4.5**. As depicted, the model consisted of two main components: 1) a generator (G) for creating the material segmentation map from the punch force (c; a scalar value) and a latent variable (z; a vector); 2) a discriminator (D) for discriminating the generated segmentation map ($G(c, z)$) from the ground truth (x). The random latent variable z was introduced to avoid deterministic mapping, considering that the SPR experiment was repeated multiple times per punch force (*i.e.*, there were three and two OM images per punch force for the CFRP-GA590DP and SPFC590DP-Al5052 combinations, respectively). In the cGAN, the latent variable

allowed us to obtain slightly different results while maintaining the main characteristics from the given condition (*c*), which would be the major geometrical properties such as head height, interlock length, and bottom thickness in SPR.

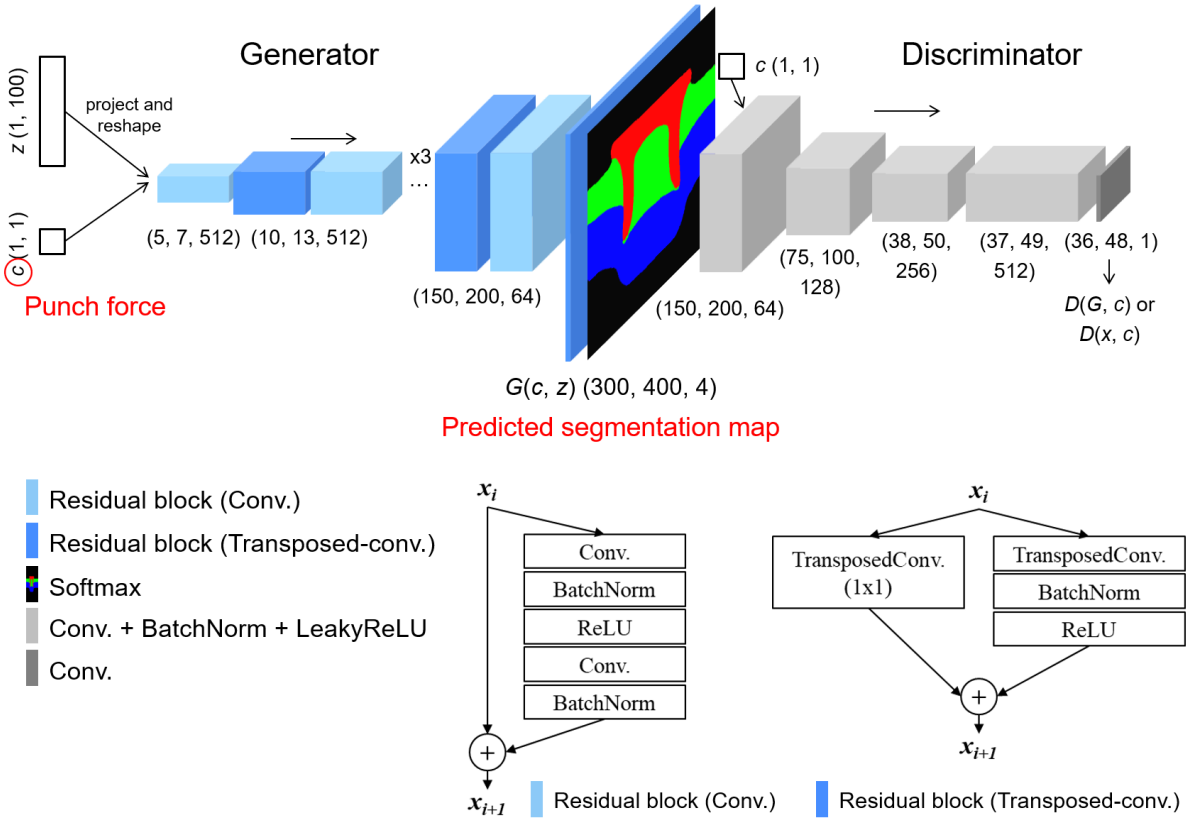

Figure 4.6. Detailed architecture of the scalar-to-segmentation generator. It was based on the CNN and cGAN structures with residual blocks. Presented segmentation map and spatial dimensions of the inner layers are for the CFRP-GA590DP material combination type.

In **Figure 4.6**, four kinds of convolutional blocks are presented and the numbers below each block denote the shape of the convoluted images (H, W, C; only the case of CFRP-GA590DP was presented for brevity). The *z* and *c* were input to the generator after a linear layer, and the number of initial feature maps was set to 465 and 47, respectively. (*z*: (N, 100)→reshape(linear(*z*)): (N, 5, 7, 465); *c*: (N, 1)→reshape(linear(*c*)): (N, 5, 7, 47); input to the generator *z* | *c*: (N, 5, 7, 512) (concatenated); N: batch size) Note that the ratio of initial feature maps was near 10:1 (465:47) for *z* and *c*, which was believed to be the optimum for our dataset (identified from trial and errors).

For the CFRP-GA590DP type, the spatial dimension of the generation was H = 300 and W = 400 as shown in the figure, and for the SPFC590DP-Al5052 type, it was H = 245 and W = 480 (accordingly, shape of the input to the generator $z \mid c$ was (N, 3, 8, 512)). Inside the generator, considering the large size of the segmentation map (300 × 400 and 240 × 480), residual connections [80, 81] were utilized as exhibited by light blue and dark blue blocks in **Figure 4.6** along with their configurations. Note that the residual connection has been widely used for super-resolution problems in computer vision[40, 41, 83-85] because it can deepen the layer while maintaining the backpropagation gradients. The residual connection adopted in the light blue block in **Figure 4.6** is the one [81] that used in pix2pixHD [40] and super-resolution GAN [41], which was composed of two convolutional layers and batch normalization layers [62]. It is worth noting that we also tested instance normalization [87] instead of batch normalization which had been adopted in pix2pixHD, however, could not improve the generation quality. The dark blue residual block in **Figure 4.6** was comprised of transposed-convolutional layers, which upscaled the input image. For an intermediate nonlinear activation for both blocks, a rectified linear unit (ReLU) [63] function was used.

The light gray and dark gray blocks used in the discriminator were a convolution–batch normalization–leaky rectified linear unit (LReLU; max(0.2x, x)) [64] layer and a single convolutional layer, respectively, which followed the well-known discriminator structure [37, 38].

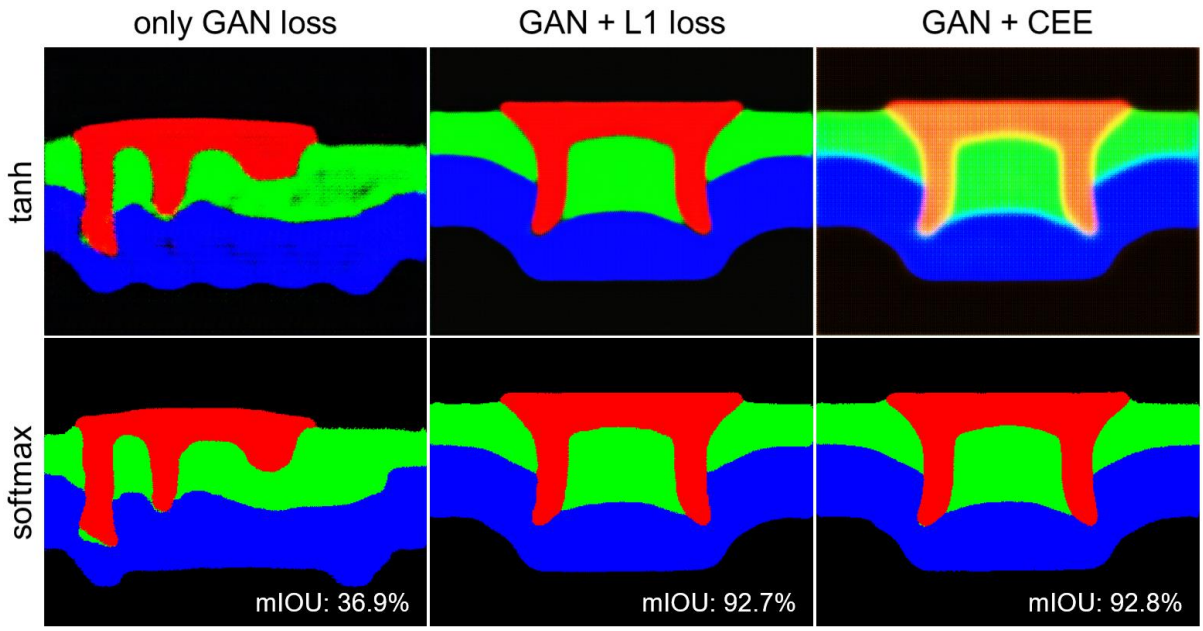

Figure 4.7. Results of the case study. We tested three objective functions (row) and two last activation functions of the generator (column). Tested sheet combination was the CFRP-GA590DP and punch force was 55 kN.

In this study, based on the results of the previous studies [38, 41, 104] that adding a conventional loss function to the GAN loss highly improved the generation quality, a case study for the objective function was conducted, considering that data type is the segmentation map. Accordingly, last activation function of the generator was also included in the case study and its results are shown in **Figure 4.7**. As seen, for the traditional loss function, L1 cost and cross entropy error (CEE) were tested (L2 was excluded from the case study, as it was known to produce a blurrier image than L1 [38]). The generator now took on the task of minimizing the added cost function as well as deceiving the discriminator. For the last activation function, hyperbolic tangent (tanh) [37] and softmax were tested. Note that for the GAN loss, we used a least-squares GAN loss (LSGAN; [89]), as it had higher stability than the traditional GAN:

$$\min_D V_D = \frac{1}{2}\mathbb{E}_{x,c}[(D(x,c)-1)^2] + \frac{1}{2}\mathbb{E}_{c,z}[(D(G(c,z),c))^2], \tag{4.1}$$

$$\min_G V_G = \frac{1}{2}\mathbb{E}_{c,z}[(D(G(c,z),c)-1)^2] + \lambda\mathcal{L}(G), \tag{4.2}$$

$$\begin{aligned}&\mathcal{L}(G) = \mathcal{L}_{L1}(G) = \mathbb{E}_{x,c,z}[\|x - G(c,z)\|_1] \text{ or} \\ &\mathcal{L}(G) = \mathcal{L}_{\text{cross entropy}}(G) = \mathbb{E}_{x,c,z}[-x\ln G(c,z)]\end{aligned}, \tag{4.3}$$

where V_D and V_G denote the objective functions for the discriminator and generator, respectively, and $\mathcal{L}(G)$ is the to-be-added conventional cost function (L1 or CEE). Further details of the GAN loss can be referred from [36, 52] and LSGAN papers [89] (not presented here for the sake of conciseness). As shown in **Figure 4.7**, without the extra loss (*i.e.* only GAN loss), the generation quality was poor and the generator failed to learn the right mapping between the input (c, z) space and the ground truth. When checking the generated images in every epoch with a loss profile, we could confirm that the initial training of the generator was not properly provoked, thus the entire training iterations were substantially unstable. However, when trained with the L1 loss (second column in **Figure 4.7**), the generator created considerably better results, and the last activation of softmax layer showed much better quality than that of tanh (sharper edges without blurring; mIOU=92.7%), which actually is an obvious result as the data type was the segmentation map (0 or 1 is allocated in each material class). With the CEE (third column in **Figure 4.7**), the generation result with the softmax layer was significantly better (mIOU=92.8%) than with the tanh layer which showed underdeveloped generation. In summary, adding conventional loss function to the GAN cost largely contributed to initiating the generator training, and given the type

of ground truth (segmentation map) it was good to use the softmax layer as the last activation function of the generator. Moreover, both GAN + L1 and GAN + CEE worked fine, however, GAN + CEE was adopted in this study as it showed a slightly higher accuracy with clearer material boundaries.

4.3.3 Training details

Material segmentation model was trained on dual TITAN RTX GPUs. Among a total of 41 pairs of [OM, segmentation] data (27 pairs for CFRP-GA590DP and 14 pairs for SPFC590DP-Al5052), 7, 7, and 27 pairs were used for the model validation, test, and training, respectively (randomly chosen). Note here, both CFRP-GA590DP and SPFC590DP-Al5052 sheet combinations were included in the training dataset, to let the AI model segment the OM images regardless of the material types. The training data were augmented to 5400 (27 × 200) in every epoch by applying random cropping, random scaling, and random flipping of the images. The model was trained according to the training protocol of DeepLabv3+, which was a polynomial learning rate policy where the initial learning rate was decayed by $(1 - \text{iteration}/\text{iteration}_{max})^{power}$ (power was set to 0.9). Two training sessions were conducted successively: in the first 100 epochs, the initial learning rate was 0.001 with batch size of 15 and output stride of 16, and in the next 100 epochs, the initial learning rate was 0.0001 with batch size of 5 and output stride of 8. The L2 weight decay coefficient was 10^{-7}. We trained the model from the scratch (no pretraining) and the parameters in batch norm layers were not frozen even when the output stride was 8. For a segmentation performance measure, intersection-over-union (IOU) and pixel accuracy (PA) were calculated [33].

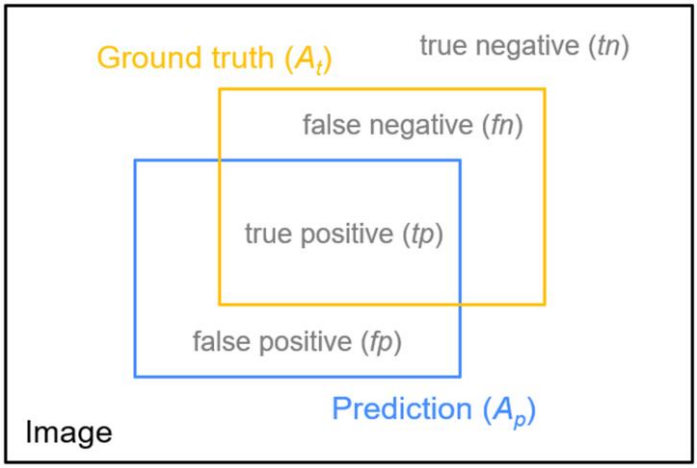

Figure 4.8. Schematic diagram for the image segmentation.

In **Figure 4.8**, a schematic showing image segmentation is presented. The blue square is the predicted segmentation area (A_p) and the orange square is the ground truth area (A_t). Note that the area encompassing the predicted values was segregated into "positive" and "negative"; the area encompassing the correct predictions was marked as "true" and that encompassing the incorrect predictions was marked as "false". From the figure, IOU and PA were calculated as

$$\mathrm{IOU} = \frac{tp}{tp + fp + fn} = \frac{A_p \cap A_t}{A_p \cup A_t}, \tag{4.4}$$

$$\mathrm{PA} = \frac{tp + tn}{tp + tn + fp + fn} . \tag{4.5}$$

The PA was defined as the number of correctly classified pixels over the total number of pixels. The IOUs and PAs of all classes (in SPR, four classes exist: rivet, upper sheet, lower sheet, and background) were averaged (namely, mIOU and mPA, respectively; 'm' denotes mean) and used for the performance measure in material segmentation.

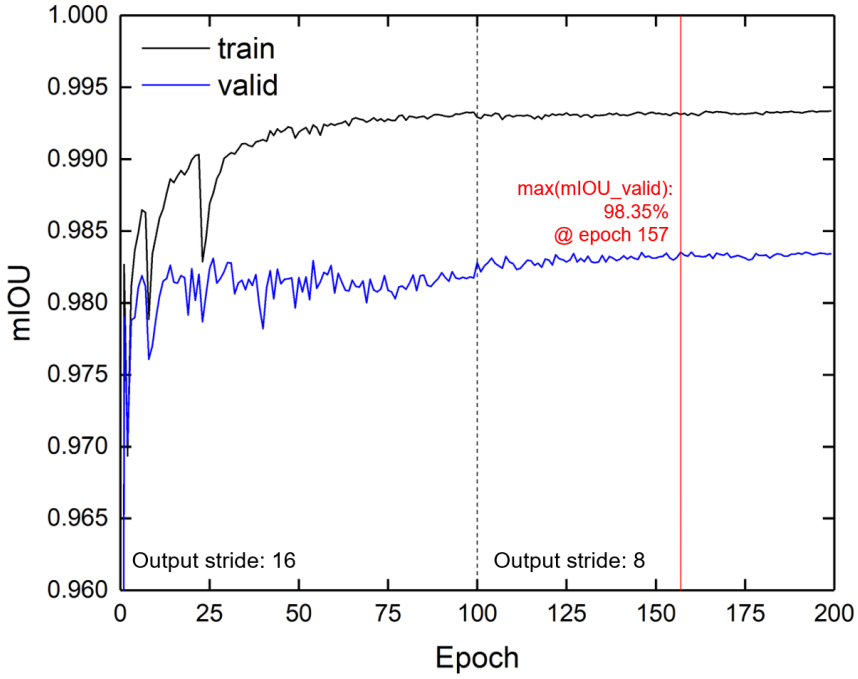

Figure 4.9. mIOU-epoch profile for the training (black) and validation (blue) sets (DeepLabv3+).

In **Figure 4.9**, the mIOU-epoch curves for the training dataset (black line) and the validation dataset (blue line) were presented, for the two successive training sessions (output stride=16 in epoch 0~99 and 8 in epoch 100~199). The validation accuracy reached the maximum at epoch 157 (98.35%), so the model parameters at this epoch were used for testing the model (*i.e.* the model was validated at epoch 157). In a test phase, three types of test sets comprising center-cropped CFRP-GA590DP OM images of 300 × 400 pixels (test set 1), full-scale CFRP-GA590DP OM images of 480 × 640 pixels (test set 2), and SPFC590DP-Al5052 OM images of 300 × 400 pixels (test set 3) were used. Note that the test set 2 (full-scale) was introduced to simply confirm the general performance of the segmentation model, and the material segmentation results for the test set 1 (center-cropped) and 3 were the data to be used in the next cross-section predictive model. It was because in the case of CFRP-GA590DP, there was no problem in measuring head height even if only the center was predicted since there was no flection at both ends, but in the case of SPFC590DP-Al5052, the flection at both ends was large so the prediction had to be made up to this area (see **Figure 4.2**). The test results are presented as the segmentation result in **chapter 4.4.1**.

The cross-section predictive model (scalar-to-seg generator) was trained on dual GTX 1080 Ti GPUs. The height and width of the convolutional weights were 4, and they were initialized from the Gaussian distribution with a standard deviation of 0.02 [37]. The training data were augmented by applying weak image transformation (× 50) and flipping the images left and right (× 2). The learning rate and lambda of the L2 weight decay were respectively 3.162×10^{-7} and 10^{-6}, respectively, which were identified using Bayesian optimization [105, 106]. The batch size was 12 and the lambda in the generator objective was 100. For the CFRP-GA590DP combination, one test force was fixed and the remaining eight punch forces were used for training. This process was repeated for all punch forces (29, 34, 39, 44, 49.8, 55, 60, 65, and 70 kN). Thus, the training was performed over nine sessions. (The model was considered validated given all nine trained models made accurate predictions.) For the SPFC590DP-Al5052 sheets, the whole data was divided into training, validation, and test sets, and four training sessions were made: a) use 40 kN as the validation set, 34 kN as the test set; b) 34 kN as the validation set, 40 kN as the test set; c) 43 kN as the validation set, 37 kN as the test set; d) 37 kN as the validation set, 43 kN as the test set. In each session, the remaining forces were used in the training. In the testing phase, after the completion of training, three fixed latent variables were input with the punch force so that three predictions were obtained. Adam [65] was adopted as the optimizer in both segmentation and predictive deep learning models.

4.4 Results and Discussion

4.4.1 Material segmentation result

Figure 4.10, **Figure 4.11**, and **Figure 4.12** present the material segmentation results for the three test sets. Each row in the figures corresponds to the test OM inputs with the punch forces of 44, 55, 65, and 70 kN for CFRP-GA590DP (**Figure 4.10** and **Figure 4.11**) and 34, 40 and 43 kN for SPFC590DP-Al5052 (**Figure 4.12**). The OM inputs, their ground truths (which were manually segmented by the authors), and AI prediction results are presented in the first, second, and third columns, respectively, and the last three columns are the segmentation images separated by channel (rivet, top sheet, and bottom sheet classes), comprising the ground truths (top) and the predictions (bottom). The prediction accuracies indicated by mIOU and mPA are displayed at the bottom-right corner in each prediction map.

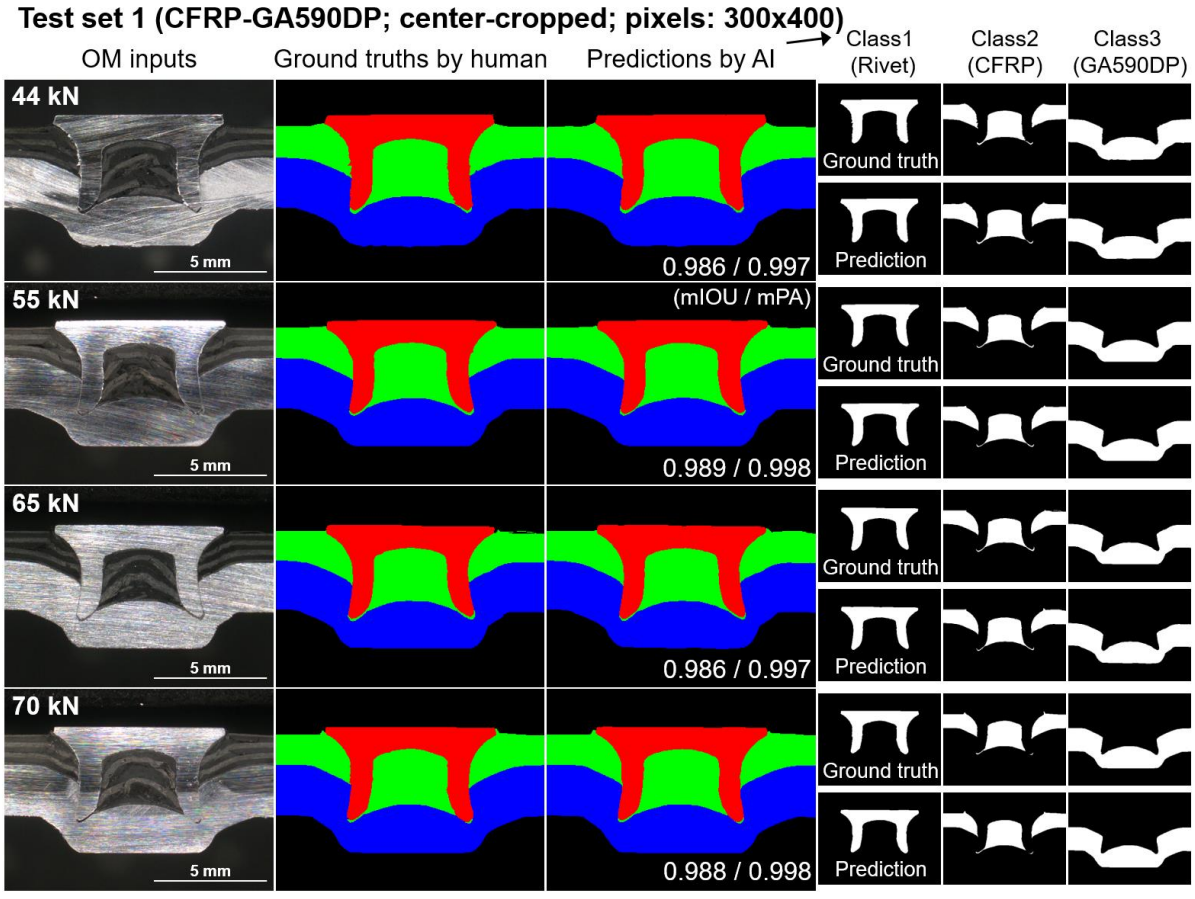

Figure 4.10. Material segmentation results for the first test set. OM inputs, their ground truths, prediction results, and channel-by-channel segmentation results for three material classes are shown in each column from left to right (the background class was omitted).

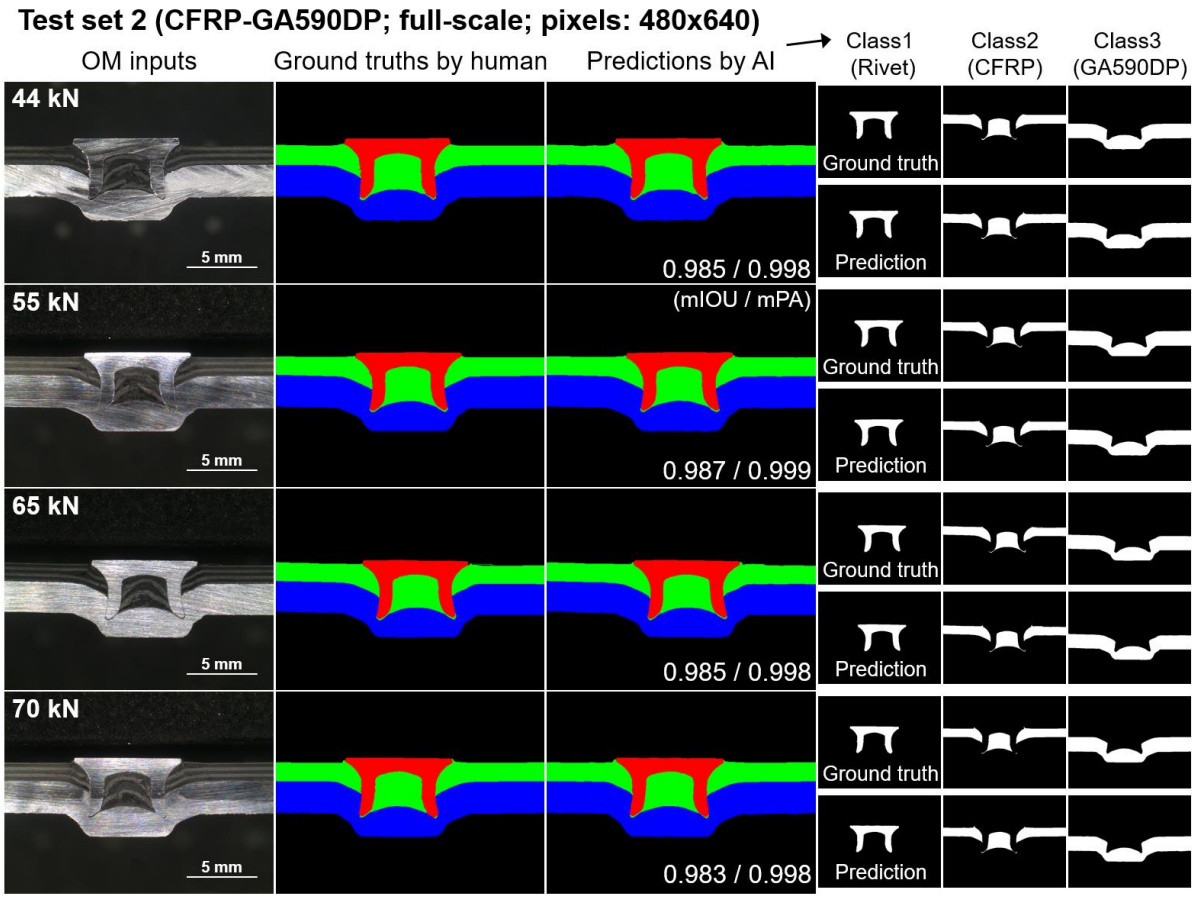

Figure 4.11. Material segmentation results for the second test set. OM inputs, their ground truths, prediction results, and channel-by-channel segmentation results for three material classes are shown in each column from left to right (the background class was omitted).

As presented in the figures, the material segmentation was successfully performed for all the test sets. The accuracy reached 98~99%, which implies that the trained AI practically segmented the OM image in the same way as would be done by humans. **Table 4.2** lists the IOU and PA accuracies for the three test sets according to class. The mean accuracy for all classes and test sets was calculated as mIOU=98.50%, mPA=99.78%. Here, it is noteworthy that, as shown in the table, the mPAs for the three test sets were 99.75%, 99.83% and 99.78%, which means that only 300 (0.25%), 522 (0.17%) and 264 (0.22%) pixels, respectively, were misclassified for the test sets when compared with the human segmentation results. These errors mostly originated from the ambiguity at the object boundaries, where the boundary itself contained multiple pixels instead of just one. For this reason, the PASCAL Visual Object Classes (VOC) challenge [33] has the rule of discounting the border area of 5 pixels and regarding them as 'void' in the data to exclude the ambiguity when measuring the accuracies of the deep learning models. Note that this rule was not followed in this study because some interlock areas in the OM image were only a few pixels wide.

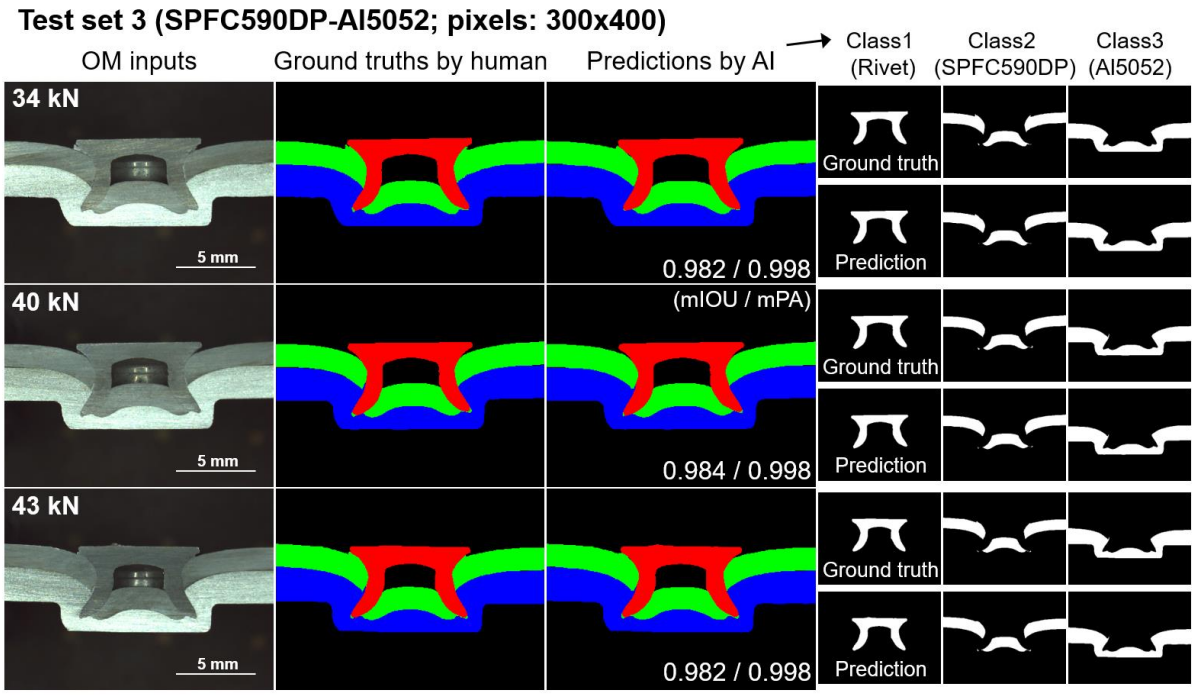

Figure 4.12. Material segmentation results for the third test set. OM inputs, their ground truths, prediction results, and channel-by-channel segmentation results for three material classes are shown in each column from left to right (the background class was omitted).

For another note, when trained with only CFRP-GA590DP data (excluding SPFC590DP-Al5052 data), about two-thirds of the training epochs was sufficient to reach a similar level of test accuracy (mIOU=98.49% and mPA=99.77% when trained with output stride of 16 for the first 50 epochs and 8 for the next 50 epochs). It was an obvious result, as the amount of to-be-trained information decreased (as can be seen in **Figure 4.11** and **Figure 4.12**, there are several clear differences between the two sheet combinations, such as existence of the hole just below the rivet). When training the model with both sheet combinations, a total of 100 epochs was not enough, and at least 150 epochs were required as presented in **chapter 4.3.3**.

After ensuring material segmentation with 98~99% accuracy, the training data to be used in the cross-section predictive model were safely replaced with the segmentation map, instead of the OM image. The cross-section predictive model can now be further trained in the future by simply inputting new OM images to the deep learning segmentation model. Also, if other sheet combinations of SPR joints are added in the future, the trained segmentation model can be easily further trained by restoring the parameters of the validated model.

Table 4.2. IOU and PA accuracies (%) for the three test sets according to the label.

1. CFRP-GA590DP (300×400)	Class1 (rivet)	Class2 (CFRP)	Class3 (GA590DP)	Class4 (background)	Mean
IOU	98.20	98.06	99.06	99.54	98.71 (mIOU)
PA	99.77	99.67	99.77	99.79	99.75 (mPA)
2. CFRP-GA590DP (480×640)	Class1 (rivet)	Class2 (CFRP)	Class3 (GA590DP)	Class4 (background)	Mean
IOU	97.67	97.87	98.75	99.76	98.51
PA	99.89	99.78	99.81	99.83	99.83
3. SPFC590DP-Al5052 (300×400)	Class1 (rivet)	Class2 (SFPC590DP)	Class3 (Al5052)	Class4 (background)	Mean
IOU	97.25	97.41	98.75	99.68	98.28
PA	99.80	99.70	99.82	99.79	99.78
mIOU avg.			-		*98.50%*
mPA avg.			-		*99.78%*

4.4.2 Cross-sectional shape prediction result

Figure 4.13 and **Figure 4.14** present the predicted material segmentation maps for each punch force in the three columns on the left (three latent variables), and their corresponding ground truths in the remaining columns on the right (repeated experiments). The random latent variables (z1, z2, z3) and the experiment repetition number (1, 2, 3 for CFRP-GA590DP and 1, 2 for SPFC590DP-Al5052) are marked at the bottom-right corner of the images. As shown, the quality of the generated (predicted) images was high without any repeated or distorted, blurred areas, owing to the modified structure and objective function. As seen in the predicted results in **Figure 4.13**, the larger the punch force, the more the rivet is pressed and the greater is the interlock length. For a more accurate comparison with the ground truths, three geometrical key factors of head height, interlock, and bottom thickness were measured in every map and plotted versus the punch force in **Figure 4.15**, **Figure 4.16**, and **Figure 4.17**, respectively (empty circle and square: CFRP-GA590DP; filled circle and square: SPFC590DP-Al5052; black: ground truth; red: prediction). As shown in **Figure 4.13-4.17**, although slightly different cross-sections were generated in each punch force according to the introduced latent variables, the three important geometrical factors were properly mapped from the input punch force. It implies that the generator was trained at a semantic level, by spontaneously grasping the relationship between the input punch forces and the cross-sectional shapes, as the training proceeded. However, in the case of

SPFC590DP-Al5052 interlock in 34~43 kN (see **Figure 4.16**), the ground truth showed less tendency and so was the prediction result, which suggests the importance of quality and quantity of the raw data (SPR experiments for the SPFC590DP-Al5052 were repeated only twice; the deep learning model becomes solid with a large amount of data and thus a more repetition of the experiments is preferred in a future work).

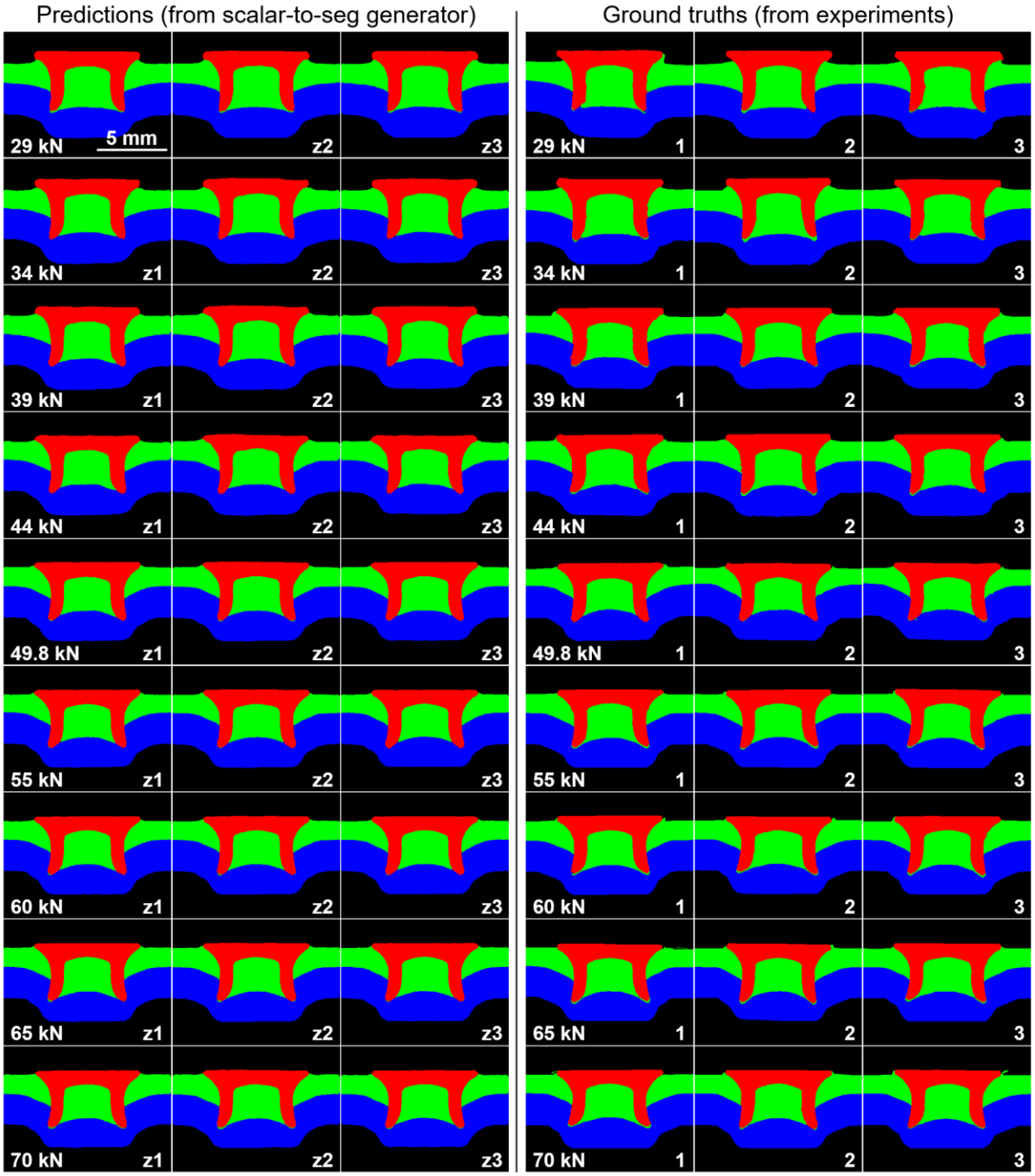

Figure 4.13. Prediction results (three columns on the left; three latent variables) and ground truths (three columns on the right; three repetitions) for the CFRP-GA590DP are presented for each punch force.

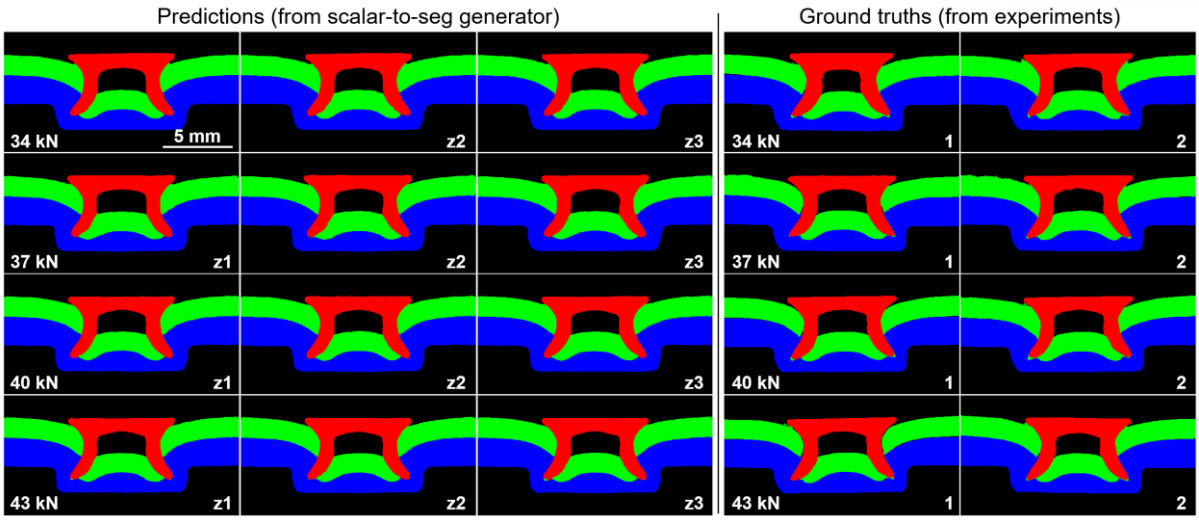

Figure 4.14. Prediction results (three columns on the left; three latent variables) and ground truths (two columns on the right; two repetitions) for the SPFC590DP-Al5052 are presented for each punch force.

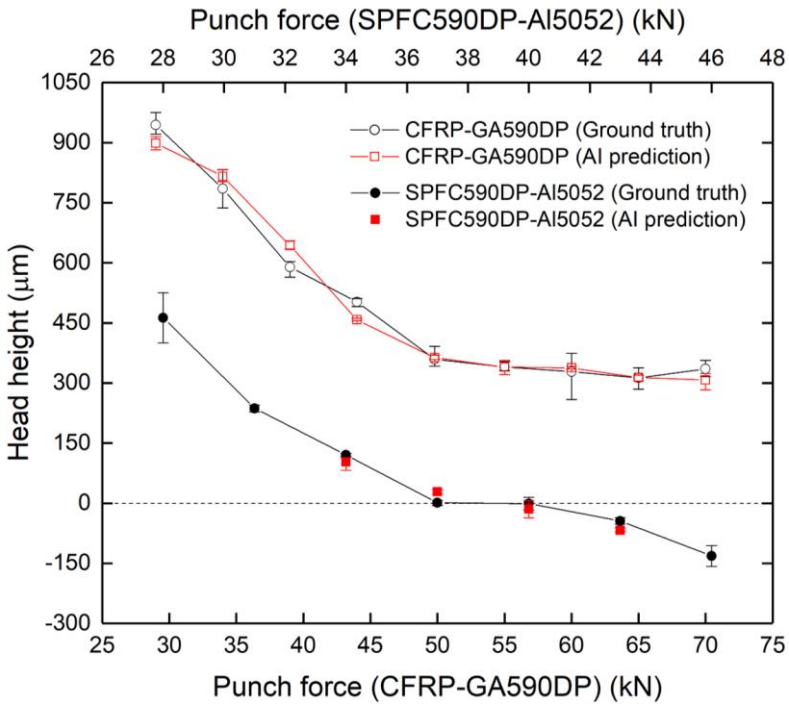

Figure 4.15. Results of the head height measurement from the AI prediction results (red) and the ground truths (black), according to the punch force (CFRP-GA590DP: left *y*–bottom *x*; SPFC590DP-Al5052: left *y*–top *x*).

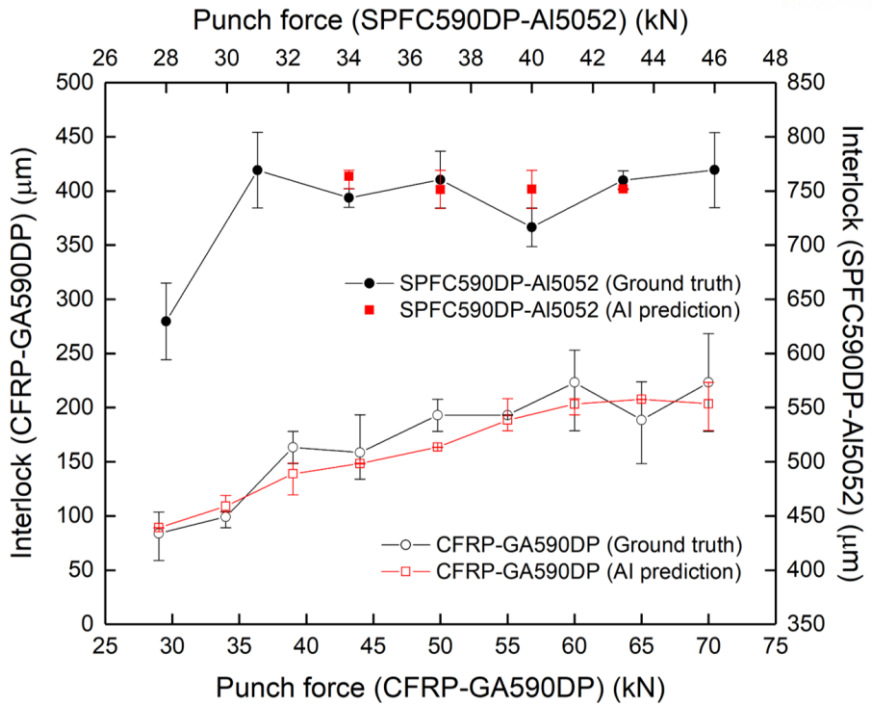

Figure 4.16. Results of the interlock length measurement from the AI prediction results (red) and the ground truths (black), according to the punch force (CFRP-GA590DP: left *y*–bottom *x*; SPFC590DP-Al5052: right *y*–top *x*).

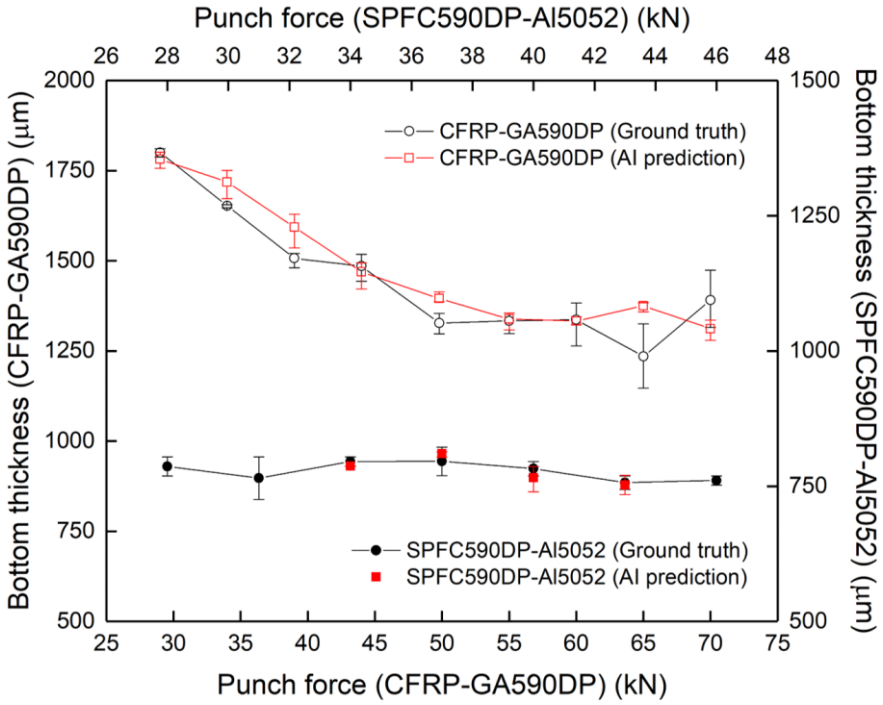

Figure 4.17. Results of the bottom thickness measurement from the AI prediction results (red) and the ground truths (black), according to the punch force (CFRP-GA590DP: left *y*–bottom *x*; SPFC590DP-Al5052: right *y*–top *x*).

Table 4.3 (for CFRP-GA590DP) and **Table 4.4** (for SPFC590DP-Al5052) present the accuracies for the head height, interlock, and bottom thickness for each punch force. For the CFRP-GA590DP, the averaged accuracy across the punch forces for the head height, interlock, and bottom thickness were 95.66%, 90.78%, and 96.16%, respectively (the average of the averages was 94.20%), and for the SPFC590DP-Al5052, they were 92.76%, 97.55%, and 98.60%, respectively (the average of the averages was 96.31%).

Table 4.3. Calculated accuracies of the head height, interlock, and bottom thickness according to the punch force for the CFRP-GA590DP sheets.

Accuracy (%)	29 kN	34 kN	39 kN	44 kN	49.8 kN
Head height	95.24	96.06	90.91	91.22	98.94
Interlock	93.72	90.16	85.09	93.61	84.74
Bottom thickness	98.97	96.01	94.29	98.96	94.90
Accuracy (%)	55 kN	60 kN	65 kN	70 kN	Average
Head height	99.88	97.03	99.88	91.75	95.66
Interlock	97.59	91.11	89.83	91.18	90.78
Bottom thickness	99.64	99.74	88.62	94.32	96.16
Mean accuracy			-		*94.20%*

Table 4.4. Calculated accuracies of the head height, interlock, and bottom thickness according to the punch force for the SPFC590DP-Al5052 sheets.

Accuracy (%)	34 kN	37 kN	40 kN	43 kN	Average
Head height	85.51	90.75	97.77	97.03	92.76
Interlock	97.34	98.78	95.12	98.95	97.55
Bottom thickness	98.95	98.22	97.88	99.37	98.60
Mean accuracy			-		*96.31%*

For more intuitive comparison, **Figure 4.18(a)** shows the AI-predicted CFRP-GA590DP cross-sectional shapes (left half) and the OM images obtained from the SPR experiments (right half) for the punch forces (29, 39, 55, and 70 kN) alongside the measurement data. (**Figure 4.18(b)**: SPFC590DP-Al5052 for the punch forces of 34, 40, and 43 kN) As seen, the region of each material segment in the AI-predicted results is almost identical to that of the OM images. The presented results suggest that the deep learning architecture can be very useful in predicting the deformed cross-sectional shape in the SPR process.

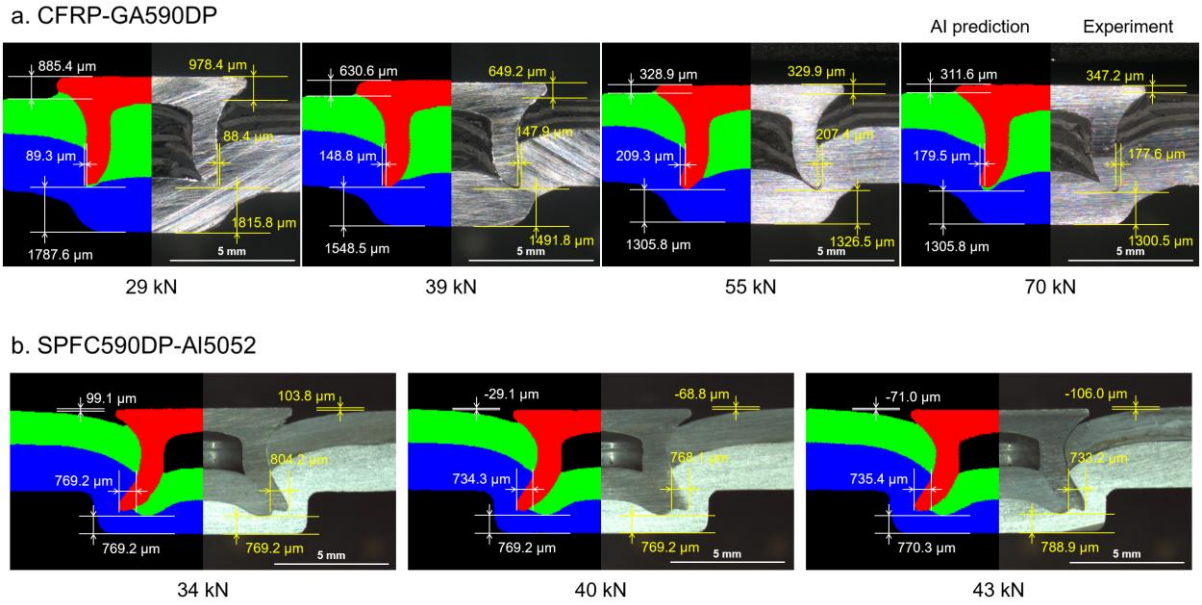

Figure 4.18. Comparison between the AI prediction results (left half) and the OM images obtained from the experiments (right half). (a) The punch forces for the CFRP-GA590DP were 29 kN, 39 kN, 55 kN, and 70 kN, respectively. (b) The punch forces for the SPFC590DP-Al5052 were 34 kN, 40 kN, and 43 kN, respectively. These forces were applied in the experiments and provided as inputs to the deep learning model as scalar values. Measurements for the head height, interlock, and bottom thickness are depicted in each figure.

4.5 Conclusion

We conducted a study for predicting the cross-sectional shapes in SPR of CFRP and GA590DP steel sheets and SPFC590DP and Al5052 sheets by applying state-of-the-art deep learning algorithms. Two deep learning architectures were trained for data characterization and prediction. The major findings can be summarized as follows:

- To characterize the cross-sectional shapes of the self-pierced specimens, the OM images obtained from the SPR experiment were segmented material by material by employing a CNN-based deep learning architecture. We obtained mIOU and mPA values of 98.50% and 99.78%, respectively.

- Using the scalar punch force, the cross-sectional shape was predicted by a generative model based on CNN and cGAN architectures with residual blocks (scalar-to-seg generator). The cost function was modified considering the type of the ground truth (segmentation map). The proposed generator can produce segmentation images of considerably better quality.

- The accuracies of the cross-sectional shape predictions of the major geometrical factors in SPR, namely, the rivet head height, interlock length, and bottom thickness, were 95.66%, 90.78%, and 96.16%, respectively (mean 94.20%) for CFRP-GA590DP, and 92.76%, 97.55%, and 98.60% (mean 96.31%) for SPFC590DP-Al5052, respectively.

Chapter 5. Summary and Future Perspectives

5.1 Summary

In this dissertation, novel applications of deep learning to the three representative modern materials processing technologies (*i.e.*, laser heat treatment, laser keyhole welding, and self-piercing riveting) are presented. The proposed models were based on the state-of-the-art deep learning algorithms and they were trained by the data obtained from experiments or simulations. The prediction accuracy for the three processing techniques was all good, which demonstrates the potential for deep learning very well. As presented in this dissertation so far, there are three main advantages of applying a deep learning architecture to the materials processing technology.

- Compared with the conventional physics-based methods of numerically solving the differential equations, the deep learning method can predict the process without any assumptions (data-driven), thus prediction results that are close to the reality can be achieved. Therefore, a well-trained deep learning model generally reveals a very good accuracy.

- Owing to the numerous trainable parameters in the deep neural layers, very complicated problems can be effectively predicted, as long as a sufficient amount of data is given.

- Compared to the traditional prediction method (*e.g.* FEM simulation), the predicting time is unimaginably short (within a few seconds), once the training and validation are finished. Moreover, meshing and time step are totally nonfactors.

To conclude, throughout the dissertation, we demonstrated novel deep learning applications to the three materials processing technologies, with detailed explanations on the theoretical and technical aspects of the developed models. We expect that the presented frameworks can greatly contribute to initiating the establishment of AI-based process control system in future advanced manufacturing technologies, along with the fourth industrial (superintelligence) revolution.

5.2 Future Perspectives

For a future deep learning-based study, several appreciable aspects regarding the three materials processing techniques are suggested in this chapter, for more developed and advanced deep learning applications.

First, we suggest raw data extension for all deep learning models presented in this dissertation, as it is the surest way to construct a more accurate and robust AI model. The more training data, the more the AI model is exposed to diverse environments and experiences, which strengthens the mapping between the inputs and predictions. As a result, AI models trained with a larger amount of data can cope with a much wider variety of situations, which is especially important for high-precision processing techniques.

Second, in laser heat treatment, we suggest building a deep learning model for the prediction of deformed shape, not only the hardness distribution. As a technique for enhancing surface properties of dies such as strength, hardness, and wear resistance, laser heat treatment has been widely employed, however, due to the strict requirement of dies (high precision), minimizing thermal deformation as well as attaining the desired hardness has been a challenging issue. According to the authors' previous study [1], the deflection angle has a strong correlation with the cooling time, which can be calculated from the temperature profile. Therefore, by modifying the existing model (presented in this dissertation), it would be possible to construct a deep learning model that predicts the hardness distribution on a deformed cross-section, from the temperature distribution.

Third, in laser keyhole welding, it is recommended enlarging the resolution of generated images by stacking additional enhancers in the second generator, as the typical optical microscopic resolution is higher than that of used in this dissertation. In addition, realization of the detailed microstructures can be expected by increasing the resolution. It is also worth studying the effect of keyhole fluctuation, which causes slightly changing bead shape under the same laser processing parameters. The results in this dissertation indicated that the employed data augmentation worked well, but more fundamental research on this topic could be done in a subsequent study.

Lastly, in laser keyhole welding and self-piercing riveting, there are other subsidiary process variables affecting the weld bead and SPR joint shape, such as ambient pressure and shielding gas flow (laser welding; [107-110]), and geometrical shape and dimension of the rivet, sheets, and die, material properties, and punch speed (SPR; [26]). So, in the following study, they can be additionally introduced to the generator with the original input variables (*i.e.* laser intensity and beam scanning speed in laser welding, and punch force in SPR).

References

[1] S. Oh and H. Ki, "Prediction of hardness and deformation using a 3-D thermal analysis in laser hardening of AISI H13 tool steel," (in English), *Appl Therm Eng,* vol. 121, pp. 951-962, Jul 5 2017, doi: 10.1016/j.applthermaleng.2017.04.156.

[2] S. Oh and H. Ki, "Deep learning model for predicting hardness distribution in laser heat treatment of AISI H13 tool steel," (in English), *Appl Therm Eng,* vol. 153, pp. 583-595, May 5 2019, doi: 10.1016/j.applthermaleng.2019.01.050.

[3] S. Oh and H. Ki, "Cross-Section Bead Image Prediction in Laser Keyhole Welding of AISI 1020 Steel Using Deep Learning Architectures," *IEEE Access,* vol. 8, pp. 73359-73372, 2020.

[4] D. Von der Linde, K. Sokolowski-Tinten, and J. Bialkowski, "Laser–solid interaction in the femtosecond time regime," *Appl Surf Sci,* vol. 109, pp. 1-10, 1997.

[5] K. C. Phillips, H. H. Gandhi, E. Mazur, and S. K. Sundaram, "Ultrafast laser processing of materials: a review," (in English), *Adv Opt Photonics,* vol. 7, no. 4, pp. 684-712, Dec 31 2015, doi: 10.1364/Aop.7.000684.

[6] J. D. Majumdar and I. Manna, "Laser processing of materials," *Sadhana,* vol. 28, no. 3-4, pp. 495-562, 2003.

[7] W. M. Steen and J. Mazumder, *Laser material processing*. springer science & business media, 2010.

[8] J. Mazumder, "Laser Heat-Treatment - the State of the Art," (in English), *Jom-J Min Met Mat S,* vol. 35, no. 5, pp. 18-26, 1983. [Online]. Available: <Go to ISI>://WOS:A1983QR64400001.

[9] H. Ki and S. So, "Process map for laser heat treatment of carbon steels," (in English), *Opt Laser Technol,* vol. 44, no. 7, pp. 2106-2114, Oct 2012, doi: 10.1016/j.optlastec.2012.03.018.

[10] S. So and H. Ki, "Effect of specimen thickness on heat treatability in laser transformation hardening," (in English), *Int J Heat Mass Tran,* vol. 61, pp. 266-276, Jun 2013, doi: 10.1016/j.ijheatmasstransfer.2013.01.048.

[11] H. Ki, S. So, and S. Kim, "Laser transformation hardening of carbon steel sheets using a heat sink," (in English), *J Mater Process Tech,* vol. 214, no. 11, pp. 2693-2705, Nov 2014, doi: 10.1016/j.jmatprotec.2014.06.013.

[12] S. Kim, S. So, and H. Ki, "Controlling thermal deformation using a heat sink in laser transformation hardening of steel sheets," (in English), *J Mater Process Tech,* vol. 216, pp. 455-462, Feb 2015, doi: 10.1016/j.jmatprotec.2014.10.014.

[13] J. Agren, "A Revised Expression for the Diffusivity of Carbon in Binary Fe-C Austenite," (in English), *Scripta Metall Mater,* vol. 20, no. 11, pp. 1507-1510, Nov 1986, doi: Doi 10.1016/0036-9748(86)90384-4.

[14] G. Krauss, "Steels: processing, structure, and performance.," *USA: ASM International,* 2005.

[15] *http://matweb.com*, 2017.

[16] B. Ribic, T. A. Palmer, and T. DebRoy, "Problems and issues in laser-arc hybrid welding," (in English), *Int Mater Rev,* vol. 54, no. 4, pp. 223-244, Jul 2009, doi: 10.1179/174328009x411163.

[17] S. Katayama, *Handbook of laser welding technologies*. Elsevier, 2013.

[18] J. Kim, S. Oh, and H. Ki, "A study of keyhole geometry in laser welding of zinc-coated and uncoated steels using a coaxial observation method," (in English), *J Mater Process Tech,* vol. 225, pp. 451-462, Nov 2015, doi: 10.1016/j.jmatprotec.2015.06.029.

[19] H. Ki, P. S. Mohanty, and J. Mazumder, "Modeling of laser keyhole welding: Part I. Mathematical modeling, numerical methodology, role of recoil pressure, multiple reflections, and free surface evolution," (in English), *Metall Mater Trans A,* vol. 33, no. 6, pp. 1817-1830, Jun 2002, doi: DOI 10.1007/s11661-002-0190-6.

[20] H. Ki, P. S. Mohanty, and J. Mazumder, "Modeling of laser keyhole welding: Part II. Simulation of keyhole evolution, velocity, temperature profile, and experimental verification," (in English), *Metall Mater Trans A,* vol. 33, no. 6, pp. 1831-1842, Jun 2002, doi: DOI 10.1007/s11661-002-0191-5.

[21] M. Courtois, M. Carin, P. L. Masson, S. Gaied, and M. Balabane, "A new approach to compute multi-reflections of laser beam in a keyhole for heat transfer and fluid flow modelling in laser

welding," *Journal of Physics D: Applied Physics,* vol. 46, no. 50, 2013, doi: 10.1088/0022-3727/46/50/505305.
[22] https://en.wikipedia.org/wiki/Rivet, May 2020.
[23] P. Briskham, N. Blundell, L. Han, R. Hewitt, K. Young, and D. Boomer, "Comparison of self-pierce riveting, resistance spot welding and spot friction joining for aluminium automotive sheet," SAE Technical Paper, 0148-7191, 2006.
[24] X. He, C. Deng, and X. Zhang, "Fretting behavior of SPR joining dissimilar sheets of titanium and copper alloys," *Metals,* vol. 6, no. 12, p. 312, 2016.
[25] S. Blacket, "Self-pierce riveting technology. A joining solution for light metals," in *Proceedings of the 1st International Light Metals Technology Conference, Brisbane, Qld, Australia*, 2003, pp. 383-388.
[26] R. Haque, "Quality of self-piercing riveting (SPR) joints from cross-sectional perspective: A review," (in English), *Arch Civ Mech Eng,* vol. 18, no. 1, pp. 83-93, Jan 2018, doi: 10.1016/j.acme.2017.06.003.
[27] W. Voelkner, "Present and future developments of metal forming: selected examples," (in English), *J Mater Process Tech,* vol. 106, no. 1-3, pp. 236-242, Oct 31 2000, doi: Doi 10.1016/S0924-0136(00)00620-8.
[28] M. Easton *et al.*, "Magnesium alloy applications in automotive structures," (in English), *Jom-Us,* vol. 60, no. 11, pp. 57-62, Nov 2008, doi: 10.1007/s11837-008-0150-8.
[29] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing letters,* vol. 3, pp. 18-23, 2015.
[30] D. Poole, A. Mackworth, and R. Goebel, "Computational Intelligence," 1998.
[31] T. M. Mitchell, "Machine learning," ed: McGraw-hill New York, 1997.
[32] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and trends in signal processing,* vol. 7, no. 3–4, pp. 197-387, 2014.
[33] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision,* vol. 88, no. 2, pp. 303-338, 2010.
[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009: Ieee, pp. 248-255.
[35] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," (in English), *P Ieee,* vol. 86, no. 11, pp. 2278-2324, Nov 1998, doi: Doi 10.1109/5.726791.
[36] I. J. Goodfellow *et al.*, "Generative adversarial nets," *Advances in Neural Information Processing Systems,* vol. 27, pp. 2672-2680, 2014.
[37] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434,* 2015.
[38] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Computer Vision and Pattern Recognition (CVPR),* 2017.
[39] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *Proceedings of the IEEE International Conference on Computer Vision (ICCV),* 2017.
[40] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798-8807.
[41] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681-4690.
[42] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9459-9468.
[43] S. Kou, D. K. Sun, and Y. P. Le, "A Fundamental-Study of Laser Transformation Hardening," (in English), *Metall Trans A,* vol. 14, no. 4, pp. 643-653, 1983, doi: Doi 10.1007/Bf02643780.

[44] J. Meijer and I. van Sprang, "Optimization of laser beam transformation hardening by one single parameter," *Annals of the CIRP,* vol. 40, pp. 183-186, 1991.
[45] R. Komanduri and Z. B. Hou, "Thermal analysis of the laser surface transformation hardening process," (in English), *Int J Heat Mass Tran,* vol. 44, no. 15, pp. 2845-2862, Aug 2001, doi: Doi 10.1016/S0017-9310(00)00316-1.
[46] M. K. H. Leung, H. C. Man, and J. K. Yu, "Theoretical and experimental studies on laser transformation hardening of steel by customized beam," (in English), *Int J Heat Mass Tran,* vol. 50, no. 23-24, pp. 4600-4606, Nov 2007, doi: 10.1016/j.ijheatmasstransfer.2007.03.022.
[47] S. Martinez, A. Lamikiz, E. Ukar, I. Tabernero, and I. Arrizubieta, "Control loop tuning by thermal simulation applied to the laser transformation hardening with scanning optics process," (in English), *Appl Therm Eng,* vol. 98, pp. 49-60, Apr 5 2016, doi: 10.1016/j.applthermaleng.2015.12.037.
[48] J. Günther, P. M. Pilarski, G. Helfrich, H. Shen, and K. Diepold, "First Steps Towards an Intelligent Laser Welding Architecture Using Deep Neural Networks and Reinforcement Learning," *Procedia Technology,* vol. 15, pp. 474-483, 2014, doi: 10.1016/j.protcy.2014.09.007.
[49] F. Lambiase, A. M. Di Ilio, and A. Paoletti, "Prediction of laser hardening by means of neural network," *Procedia CIRP,* vol. 12, pp. 181-186, Oct 2013, doi: 10.1016/j.procir.2013.09.032.
[50] M. Babic *et al.*, "Prediction of the hardness of hardened specimens with a neural network," *Materials and technology,* vol. 48, no. 3, pp. 409-414, 2014.
[51] A. G. Jerniti, A. E. Ouafi, and N. Barka, "A Predictive Modeling Based on Regression and Artificial Neural Network Analysis of Laser Transformation Hardening for Cylindrical Steel Workpieces," *Journal of Surface Engineered Materials and Advanced Technology,* vol. 06, no. 04, pp. 149-163, 2016, doi: 10.4236/jsemat.2016.64014.
[52] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784,* 2014.
[53] H. Pantsar and V. Kujanpaa, "Diode laser beam absorption in laser transformation hardening of low alloy steel," (in English), *J Laser Appl,* vol. 16, no. 3, pp. 147-153, Aug 2004, doi: Doi 10.2351/1.1710879.
[54] *https://www.engineeringtoolbox.com*, 2016.
[55] *https://www.tensorflow.org/tutorials/images/deep_cnn#convolutional-neuralnetworks*, May 2020.
[56] I. J. Goodfellow, "NIPS 2016 Tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160,* 2017.
[57] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 2528-2535, 2010.
[58] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. Lecture Notes in Computer Science. Springer, Cham.,* vol. 8689, pp. 818-833., 2014.
[59] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 5188-5196, 2015.
[60] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *MICCAI,* pp. 234-241, 2015.
[61] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," *European Conference on Computer Vision (ECCV),* pp. 702-716, 2016.
[62] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167,* 2015.
[63] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proceedings of the 27th International Conference on Machine Learning,* pp. 807-814, 2010.
[64] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, 2013, vol. 30, no. 1, p. 3.
[65] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[66] A. Squillace, U. Prisco, S. Ciliberto, and A. Astarita, "Effect of welding parameters on morphology and mechanical properties of Ti-6Al-4V laser beam welded butt joints," (in English), *J Mater Process Tech,* vol. 212, no. 2, pp. 427-436, Feb 2012, doi: 10.1016/j.jmatprotec.2011.10.005.

[67] K. N. Lankalapalli, J. F. Tu, and M. Gartner, "A model for estimating penetration depth of laser welding processes," (in English), *J Phys D Appl Phys,* vol. 29, no. 7, pp. 1831-1841, Jul 14 1996, doi: Doi 10.1088/0022-3727/29/7/018.

[68] C. Lampa, A. F. H. Kaplan, J. Powell, and C. Magnusson, "An analytical thermodynamic model of laser welding," (in English), *J Phys D Appl Phys,* vol. 30, no. 9, pp. 1293-1299, May 7 1997, doi: Doi 10.1088/0022-3727/30/9/004.

[69] W. S. Chang and S. J. Na, "A study on the prediction of the laser weld shape with varying heat source equations and the thermal distortion of a small structure in micro-joining," (in English), *J Mater Process Tech,* vol. 120, no. 1-3, pp. 208-214, Jan 15 2002, doi: Doi 10.1016/S0924-0136(01)00716-6.

[70] K. Y. Benyounis, A. G. Olabi, and M. S. J. Hashmi, "Effect of laser welding parameters on the heat input and weld-bead profile," *J Mater Process Tech,* vol. 164-165, pp. 978-985, 2005, doi: 10.1016/j.jmatprotec.2005.02.060.

[71] D. B. Hann, J. Iammi, and J. Folkes, "A simple methodology for predicting laser-weld properties from material and laser parameters," *Journal of Physics D: Applied Physics,* vol. 44, no. 44, 2011, doi: 10.1088/0022-3727/44/44/445401.

[72] J. Volpp and F. Vollertsen, "Analytical Modeling of the Keyhole Including Multiple Reflections for Analysis of the Influence of Different Laser Intensity Distributions on Keyhole Geometry," *Physics Procedia,* vol. 41, pp. 460-468, 2013, doi: 10.1016/j.phpro.2013.03.102.

[73] J. Kim and H. Ki, "Scaling law for penetration depth in laser welding," (in English), *J Mater Process Tech,* vol. 214, no. 12, pp. 2908-2914, Dec 2014, doi: 10.1016/j.jmatprotec.2014.06.025.

[74] W. J. Suder and S. Williams, "Power factor model for selection of welding parameters in CW laser welding," *Optics & Laser Technology,* vol. 56, pp. 223-229, 2014, doi: 10.1016/j.optlastec.2013.08.016.

[75] R. Fabbro, "Scaling laws for the laser welding process in keyhole mode," *J Mater Process Tech,* vol. 264, pp. 346-351, 2019, doi: 10.1016/j.jmatprotec.2018.09.027.

[76] W. S. Chang and S. J. Na, "Prediction of laser-spot-weld shape by numerical analysis and neural network," (in English), *Metall Mater Trans B,* vol. 32, no. 4, pp. 723-731, Aug 2001, doi: DOI 10.1007/s11663-001-0126-3.

[77] K. R. Balasubramanian, G. Buvanashekaran, and K. Sankaranarayanasamy, "Modeling of laser beam welding of stainless steel sheet butt joint using neural networks," *CIRP Journal of Manufacturing Science and Technology,* vol. 3, no. 1, pp. 80-84, 2010, doi: 10.1016/j.cirpj.2010.07.001.

[78] L. Jacques and A. El Ouafi, "ANN Based Predictive Modelling of Weld Shape and Dimensions in Laser Welding of Galvanized Steel in Butt Joint Configurations," *Journal of Minerals and Materials Characterization and Engineering,* vol. 06, no. 03, pp. 316-332, 2018, doi: 10.4236/jmmce.2018.63022.

[79] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," (in English), *Science,* vol. 313, no. 5786, pp. 504-507, Jul 28 2006, doi: 10.1126/science.1127647.

[80] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[81] S. Gross and M. Wilber, "Training and investigating residual nets," *http://torch.ch/blog/2016/02/04/resnets.html*, 2016.

[82] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136-144.

[83] X. Wang *et al.*, "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks," in

European Conference on Computer Vision, 2018: Springer, pp. 63-79.

[84] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2472-2481.

[85] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11065-11074.

[86] J. Patterson and A. Gibson, *Deep learning: A practitioner's approach*. " O'Reilly Media, Inc.", 2017.

[87] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022,* 2016.

[88] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, 2016: Springer, pp. 630-645.

[89] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794-2802.

[90] G. Di Franco, L. Fratini, A. Pasta, and V. F. Ruisi, "On the self-piercing riveting of aluminium blanks and carbon fibre composite panels," (in English), *Int J Mater Form,* vol. 6, no. 1, pp. 137-144, Mar 2013, doi: 10.1007/s12289-011-1067-2.

[91] X. L. Zhang *et al.*, "Influence of heat treatment on fatigue performances for self-piercing riveting similar and dissimilar titanium, aluminium and copper alloys," (in English), *Mater Design,* vol. 97, pp. 108-117, May 5 2016, doi: 10.1016/j.matdes.2016.02.075.

[92] R. Porcaro, A. G. Hanssen, M. Langseth, and A. Aalberg, "Self-piercing riveting process: An experimental and numerical investigation," (in English), *J Mater Process Tech,* vol. 171, no. 1, pp. 10-20, Jan 10 2006, doi: 10.1016/j.jmatprotec.2005.05.048.

[93] G. Casalino, A. Rotondo, and A. Ludovico, "On the numerical modelling of the multiphysics self piercing riveting process based on the finite element technique," (in English), *Adv Eng Softw,* vol. 39, no. 9, pp. 787-795, Sep 2008, doi: 10.1016/j.advengsoft.2007.12.002.

[94] E. Atzeni, R. Ippolito, and L. Settineri, "Experimental and numerical appraisal of self-piercing riveting," (in English), *Cirp Ann-Manuf Techn,* vol. 58, no. 1, pp. 17-20, 2009, doi: 10.1016/j.cirp.2009.03.081.

[95] L. Huang, J. V. Lasecki, H. Guo, and X. Su, "Finite element modeling of dissimilar metal self-piercing riveting process," *SAE International Journal of Materials and Manufacturing,* vol. 7, no. 3, pp. 698-705, 2014.

[96] M. Carandente, R. J. Dashwood, I. G. Masters, and L. Han, "Improvements in numerical simulation of the SPR process using a thermo-mechanical finite element analysis," (in English), *J Mater Process Tech,* vol. 236, pp. 148-161, Oct 2016, doi: 10.1016/j.jmatprotec.2016.05.001.

[97] Y. W. Ma, Y. B. Li, W. Hu, M. Lou, and Z. Q. Lin, "Modeling of Friction Self-Piercing Riveting of Aluminum to Magnesium," (in English), *J Manuf Sci E-T Asme,* vol. 138, no. 6, Jun 2016, doi: Artn 061007 10.1115/1.4032085.

[98] F. Hönsch, J. Domitner, C. Sommitsch, B. Götzinger, and M. Kölz, "Numerical simulation and experimental validation of self-piercing riveting (SPR) of 6xxx aluminium alloys for automotive applications," in *Journal of Physics: Conference Series*, 2018, vol. 1063, no. 1: IOP Publishing, p. 012081.

[99] R. Haque, N. S. Williams, S. E. Blacket, and Y. Durandet, "A simple but effective model for characterizing SPR joints in steel sheet," (in English), *J Mater Process Tech,* vol. 223, pp. 225-231, Sep 2015, doi: 10.1016/j.jmatprotec.2015.04.006.

[100] L. Kroll, S. Mueller, R. Mauermann, and R. Gruetzner, "Strength of self-piercing riveted joints for CFRP/aluminium sheets," in *18th International conference on composite materials (ICCM/18), South Korea,(City, 2011)*, 2011, pp. 21-26.

[101] F. Hirsch, S. Muller, M. Machens, R. Staschko, N. Fuchs, and M. Kastner, "Simulation of self-piercing rivetting processes in fibre reinforced polymers: Material modelling and parameter identification," (in English), *J Mater Process Tech,* vol. 241, pp. 164-177, Mar 2017, doi: 10.1016/j.jmatprotec.2016.10.010.

[102] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801-818.

[103] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251-1258.

[104] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536-2544.

[105] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599,* 2010.

[106] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951-2959.

[107] S. Y. Pang, K. Hirano, R. Fabbro, and T. Jiang, "Explanation of penetration depth variation during laser welding under variable ambient pressure," (in English), *J Laser Appl,* vol. 27, no. 2, May 2015, doi: Unsp 022007 10.2351/1.4913455.

[108] J. Ahn, E. G. He, L. Chen, J. Dear, and C. Davies, "The effect of Ar and He shielding gas on fibre laser weld shape and microstructure in AA 2024-T3," (in English), *Journal of Manufacturing Processes,* vol. 29, pp. 62-73, Oct 2017, doi: 10.1016/j.jmapro.2017.07.011.

[109] M. Jiang, X. Chen, Y. B. Chen, and W. Tao, "Increasing keyhole stability of fiber laser welding under reduced ambient pressure," (in English), *J Mater Process Tech,* vol. 268, pp. 213-222, Jun 2019, doi: 10.1016/j.jmatprotec.2019.01.026.

[110] L. Q. Li, G. C. Peng, J. M. Wang, J. F. Gong, and S. H. Meng, "Numerical and experimental study on keyhole and melt flow dynamics during laser welding of aluminium alloys under subatmospheric pressures," (in English), *Int J Heat Mass Tran,* vol. 133, pp. 812-826, Apr 2019, doi: 10.1016/j.ijheatmasstransfer.2018.12.165.

This dissertation is dedicated to my family, Myeongseop Oh, Hye Sook Lee, and Dain Oh.

Acknowledgement

First, I would like to express my deepest appreciation to my advisor, Professor Hyungson Ki, who has fully supported and well guided my doctoral course. His academic depth and insight helped me a lot with the research I have done during my doctoral course.

I would like to appreciate the rest of dissertation committee as well: Professor Jaeseon Lee, Professor Jaesung Jang, Dr. Jun Choi, and Dr. Sangwoo So, for their thoughtful and insightful comments.

Also, I would like to thank to my lab members: Dr. Jaehun Kim, Dr. Chun Deng, Sanseo Kim, Dr. Keunhee Lee, Dr. Haram Yeo, Hyeongwon Kim, Hyung Kook Jin, Myeonggyun Son, Hyun Kyung Kim, Hojun Na, and Kimoon Nam. Thanks to their cooperation, I was able to finish my doctoral course with good memories.

Particularly, I would like to thank my colleagues and friends at the university: Hyeonho Cho, Geonwoo Han, and Hyun Gyu Hwang in the department of mechanical engineering, and Dr. Tae Kyung Lee, Jongwon Oh, Hoyoon Jung, Dohoon Kim, and JaeHo Choi in the department of chemical engineering, and Daejung Jo, Kyeongsik Kim, and Yoonseok Choi in the department of computer science. Thanks to their mental support, I was able to continue to focus on my research. Also, I would like to thank all my other good friends in Seoul and undergraduate course.

Additionally, my special thanks go to Min Hyeok Lim, Hyeon Seon Ahn, and Hyeseong Lee for their sincere encouragement, endless support, and consistent backup. They gave me a lot of encouragement and assist to continue my doctoral study.

Finally, my deepest appreciation goes to my family, Myeongseop Oh, Hye Sook Lee, and Dain Oh for their unconditional love, consistent trust, encouragement, patience, and support. Without them, my doctoral course would not be possible.

This Page Intentionally Left Blank